



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA PODNIKATELSKÁ

FACULTY OF BUSINESS AND MANAGEMENT

ÚSTAV INFORMATIKY

INSTITUTE OF INFORMATICS

Návrh dílčí části informačního systému

Design of an information system part

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Michael Koch

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Bernard Neuwirth, Ph.D., MSc

BRNO 2021

Zadání bakalářské práce

Ústav: Ústav informatiky
Student: **Michael Koch**
Studijní program: Systémové inženýrství a informatika
Studijní obor: Manažerská informatika
Vedoucí práce: **Ing. Bernard Neuwirth, Ph.D., MSc**
Akademický rok: 2020/21

Ředitel ústavu Vám v souladu se zákonem č. 111/1998 Sb., o vysokých školách ve znění pozdějších předpisů a se Studijním a zkušebním řádem VUT v Brně zadává bakalářskou práci s názvem:

Návrh dílčí části informačního systému

Charakteristika problematiky úkolu:

Úvod
Cíle práce, metody a postupy zpracování
Teoretická východiska práce
Analýza problému
Vlastní návrhy řešení
Závěr
Seznam použité literatury
Přílohy

Cíle, kterých má být dosaženo:

Cílem práce je návrh vhodné datové struktury a algoritmů zpracování pro vyhodnocování odpovědí z dotazníků v závislosti na kombinaci různých výběrových kritériích z odpovědí. Součástí řešení bude vytvoření nového modulu informačního systému ZEFIS, který pomocí metajazyka umožní expertovi definovat nedostatky podle kombinací kritérií a na základě těchto definic bude vyhodnocovat tyto nedostatky na reálných datech z dotazníků.

Základní literární prameny:

CONOLLY, Thomas, Carolyn E. BEGG a Richard HOLOWCZAK. Mistrovství - databáze: profesionální průvodce tvorbou efektivních databází. 1. vyd. Brno: Computer Press, 2009. 584 s. ISBN 978-80-2- 1-2328-7.

GILMORE, W. J. Velká kniha PHP a MySQL 5: kompendium znalostí pro začátečníky i profesionály.

1. vyd. Brno: Zoner Press, 2007. 864 s. ISBN 80-86815-53-6.

GÁLA, Libor, Jan POUR a Zuzana ŠEDIVÁ. Podniková informatika. 3. vyd. Praha: Grada Publishing, 2015. 240 s. ISBN 978-80-247-5457-4.

MOLNÁR, Zdeněk. Efektivnost informačních systémů. 2. vyd. Praha: Grada, 2001. 179 s. ISBN 80-2470-087-5.

SODOMKA, Petr a Hana KLČOVÁ. Informační systémy v podnikové praxi. 2. vyd. Brno: Computer Press, 2010. 504 s. ISBN 978-80-251-2878-7.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2020/21

V Brně dne 28.2.2021

L. S.

Mgr. Veronika Novotná, Ph.D.
ředitel

doc. Ing. Vojtěch Bartoš, Ph.D.
děka

Abstrakt

Bakalářská práce řeší návrh a realizaci nové, dílčí části informačního systému ZEFIS, který pomocí dotazníků zjišťuje informace o nedostacích informačních systémů ve firmách. Navrhovaná změna řeší dva základní problémy. Prvním je nemožnost vyhledávat odpovědi z dotazníků v závislosti na jiných odpovědích. Druhá se týká vytvoření modulu, který by mohl na základě metajazyka, ve kterém budou popsány nedostatky firmy, vyhodnotit na základě těchto pravidel, zda se ve firmě tyto nedostatky vyskytují, či nikoli. V práci je zahrnuta také analýza stávajícího systému – způsob uchovávání dat, datové modely atd. a následné zpracování navrhnutých změn.

Abstract

Bachelor thesis is solving design and realization of new an individual part of a information system for already existing ZEFIS company which uses questionnaires to find out information about the shortcomings of information systems in companies.. The proposed change focuses primarily on 2 mains problems. The first is the inability to search for answers from questionnaires according to different answers. The second concerns the creation of a module which, on the basis of a metalanguage, the shortcomings of the company will be described, could evaluate on the basis of these rules whether these shortcomings occur in the company or not. The work includes an analysis of the existing system - the method of data storage, data models, etc. and the subsequent processing of the proposed changes.

Klíčová slova

návrh IS, informační systém, metajazyk, vyhodnocování nedostatků

Key words

System design, information system, metalanguage, evaluation of deficiencies

Bibliografická citace

KOCH, Michael. Návrh dílčí části informačního systému [online]. Brno, 2021 [cit. 2021-05-11]. Dostupné z: <https://www.vutbr.cz/studenti/zav-prace/detail/133724>.

Bakalářská práce. Vysoké učení technické v Brně, Fakulta podnikatelská, Ústav informatiky. Vedoucí práce Ing. Bernard Neuwirth, Ph.D., MSc.

Čestné prohlášení

Prohlašuji, že předložená bakalářská práce je původní a zpracoval jsem ji samostatně.

Prohlašuji, že citace použitých pramenů je úplná, že jsem ve své práci neporušil autorská práva (ve smyslu Zákona č. 121/2000 Sb., o právu autorském a o právech souvisejících s právem autorským).

V Brně dne 7. května 2021

.....
podpis autora

Poděkování

Rád bych také poděkoval vedoucímu své práce, Ing, Bernardu Neuwirthovi, Ph.D., MSc za cenné rady a připomínky k práci.

OBSAH

ÚVOD	11
CÍLE PRÁCE.....	12
1 TEORETICKÁ VÝCHODISKA PRÁCE	13
1.1 Použité technologie.....	13
1.1.1 HTML.....	13
1.1.2 PHP	13
1.1.2.1 Základy	14
1.1.2.2 Proměnné.....	15
1.1.2.3 Datové typy.....	15
1.1.2.4 Podmínkové příkazy	16
1.1.2.5 Řetězce	17
1.1.2.6 Cykly	17
1.1.2.7 Funkce.....	18
1.1.2.8 Objektově orientované programování	20
1.1.3 MYSQL.....	21
1.1.4 SQL	22
1.1.4.1 Základní dotazy	22
1.1.4.2 Agregční funkce	23
1.2 Základní pojmy	24
1.2.1 Data	24
1.2.2 Informace	25
1.2.3 Databáze	25
1.2.4 Relační model.....	25
1.2.5 Primární klíč	26
1.2.6 Cizí klíč	26
1.2.7 Kardinalita vztahu.....	26
1.2.8 Entitně – relační model	27
2 ANALÝZA SOUČASNÉHO STAVU	28
2.1 Datový model.....	31
2.1.1 Tabulka Z_FIRMY.....	31
2.1.2 Tabulka Z_SYSTEMY.....	32
2.1.3 Tabulka Z_PROCESY	32
2.1.4 Tabulka Z_AUDITY	33

2.1.5	Tabulka Z_RESPONDENTI	35
2.1.6	Tabulka Z_ODPOVEDI.....	36
2.1.7	Tabulka Z_OTAZKY.....	36
2.1.8	Tabulka Z_MOZNOSTI	37
2.2	Audity	39
2.3	Dotazník	40
2.4	Výsledky	42
3	VLASTNÍ ŘEŠENÍ.....	43
3.1	Vícekritériální vyhledávání	43
3.2	Návrh Entitně-relačního modelu	44
3.3	Úprava datových struktur	45
3.3.1	Otázka s jednou možnou odpovědí	46
3.3.2	Otázky s více odpověďmi	48
3.4	Způsob vyhledávání	50
3.4.1	Funkce systému.....	51
3.5	Identifikace nedostatků	54
3.6	Datové struktury	57
3.6.1	Datový objekt Z_NEDOSTATKY	57
3.6.2	Datový objekt Z_DOPORUCENI	58
3.6.3	Datový objekt Z_NESHODY / Z_SHODY	58
3.6.4	Datový objekt Z_SCENAR.....	59
3.7	Definice metajazyka	60
3.8	Formulace nedostatku.....	62
3.8.1	Definice neshody	64
3.8.2	Definice shody	66
3.8.3	Definice významnosti neshody.....	66
3.8.4	Definice doporučení k nedostatku.....	68
3.9	Kompilace scénáře.....	68
3.10	Finální výpočet	74
4	EKONOMICKÉ ZHODNOCENÍ.....	77
4.1	Náklady	77
4.2	Přínosy.....	78
4.2.1	Přímé přínosy.....	78
4.2.2	Nepřímé přínosy	78
5	ZÁVĚR	80

SEZNAM POUŽITÝCH ZDROJŮ	81
SEZNAM POUŽITÝCH OBRÁZKŮ.....	83
SEZNAM POUŽITÝCH TABULEK	85

ÚVOD

V době rozmachu informačních technologií je nutné kráčet s rozvojem a přizpůsobit se mu. Této změně se nevyhnou ani informační systémy menších firem a podniků, aby s narůstající poptávkou po zdokonalování služeb uspokojily potřeby klientů. Ovšem tento princip má svá úskalí a tím hlavním je zvýšení chybovosti, byť v samotném informačním systému nebo v přidružených procesech, jenž se vážou ke správnému fungování systému. Ať se jedná o sebemenší chybný detail v metodice pro zálohování dat nebo nejednoznačné určení způsobu likvidace datových medií, oba zmíněné příklady, byť zní banálně, mohou představovat závažné ohrožení podniku.

Má bakalářská práce se zaměřuje na vylepšení algoritmu pro identifikaci možných podobných problémů z různých oblastí, které jsou blíže rozepsány v Analýze současné situace. Věřím, že navrhovaný systém má silný potenciál pro předcházení možných bezpečnostních incidentů a dokáže pomoci při analýze silných a slabých stránek co se týče práce s informačním systémem dané firmy.

CÍLE PRÁCE

Hlavním cílem bakalářské práce bylo rozšířit informační systém Zefis o možnost detekovat nedostatky firem, jejich informačních systémů a procesů.

Informační systém Zefis je určen k diagnostice, zda firma užívá své informační systémy správně, a nedělá chyby typu „nejsou zálohovaná data“. K zjišťování stavu používá dotazníky. V původní verzi Zefis po vyplnění dotazníků ukazoval pomocí metody HOS8 pouze úroveň informačního systému v jednotlivých oblastech ve škále 0—100 %, na základě vyhodnocení deseti otázek pro každou oblast. V případě, že všech deset otázek bylo odpovězeno správně, ve smyslu u zkoumané oblasti byla odpověď shodná s požadovanou, bylo hodnocení oblasti 100 procent. Za každou otázku, kde odpověď byla v rozporu s požadovanou, bylo hodnocení sníženo o deset procent.

V nové verzi systému Zefis došlo k změně způsobu diagnostiky. Byl dán požadavek, že nově musí systém umožňovat definovat nedostatek na základě kombinace odpovědí na libovolné otázky dotazníků. Nedostatek definuje expert, a musí mít možnost popsat tento nedostatek pomocí metajazyka mimo vlastní program tak, aby bylo možné popisy nedostatků vytvářet a měnit bez nutnosti měnit program.

Takto popsané nedostatky pak musí umět systém diagnostikovat na reálných datech, což znamená, že zjištění každého nedostatku, popsaného metajazykem, musí být zahrnuto do kódu zpracování v programu, přičemž u každého nedostatku se vyhodnocují jiné otázky v kombinaci jiných logických operátorů.

Dílním cílem práce bylo také umožnit vyhledávání odpovědí v dotaznících v závislosti na odpovědi na jiné otázky, například zjistit, u kolika malých firem v oblasti strojírenství není ustanoven manažer informačních systémů. Tento cíl vyžadoval změnu původních datových struktur a způsobu vyhledávání v databázi pomocí SQL dotazů.

Jak bude ukázáno v kapitole vlastní řešení, oba tyto cíle byly splněny.

1 TEORETICKÁ VÝCHODISKA PRÁCE

Tato část je věnována teoretickým podkladům k vytvoření práce – jaká technologie bylo nutno využít pro zdárné dokončení práce, na jakém principu funguje či pojmy, jenž je nutné si na začátku objasnit.

1.1 Použité technologie

Jak již bylo řečeno v Cíli práce, bylo nutné rozšířit již stávající systém. Z tohoto důvodu se využily už stávající technologie, na kterých systém funguje.

1.1.1 HTML

HTML neboli *Hypertext Markup Language* je značkovací jazyk pro tvorbu webových stránek, s jehož pomocí byly vytvořeny základy webové stránky. Skládá se z takzvaných „tagů“, které jsou rozděleny do dvou základní kategorií – párové a nepárové. Klíčovým znakem pro párové tagy je nutnost použít jak otevíracího tagu, tak uzavíracího. Příkladem může být `<h1>`, jenž je určen pro tvorbu nadpisu 1. úrovně. Proto aby tag mohl fungovat podle očekávání nelze zapsat samostatné `<h1>` ale je nutné ho uzavřít pomocí zavíracího tagu `</h1>`. Mezi nepárové tagy patří např. `
`, sloužící pro odřádkování, na rozdíl od párových není třeba psát `</br>`. (6,7,8,9)

1.1.2 PHP

Jedná se o programovací jazyk pro budování webových stránek. Na rozdíl od běžných programovacích jazyků neběží na klientovi, ale na serveru, kde k němu za pomoci prohlížeče přistupuje mnoho uživatelů. Pro splnění cílů bakalářské práce nelze využít pouze HTML zejména z toho důvodu, že s jeho pomocí není možné vytvářet pokročilé programátorské algoritmy a prvky, které jsou pro nutné pro vytvoření dynamické webové stránky. Proto je nutné implementovat i PHP s vysokou adaptibilitou pro složitější informační konstrukce od operací s databází po vytvoření komplexního systému registrace a přihlašování. Velké přínosy, kvůli kterým se vyplatí PHP využívat je mnoho, největším je samozřejmě to, že je zdarma. Pokud je provozováno PHP na počítači či výkonném serveru, nehraje to sebemenší roli na jeho cenu, za využívání toho jazyka nejsou žádné poplatky. Tím, že se jedná o opensource, je možné si o něm vyhledat mnoho různých informací a článků. Je velmi často využíván, takže nepůsobí příliš velké obtíže

najít experta, kdo má již řadu zkušeností a pokud by se objevila chyba v PHP části systému, rychle by ji dokázal najít a opravit. Poslední a hlavní výhodou je fakt, že tento jazyk byl už od jeho samotného vytvoření zamýšlen pro tvorbu webových stránek to nám přináší výhody v podobě funkcí specializovaných přímo na html kód.

Syntaxe jazyka je velice jednoduchá – pro psaní kódu v PHP je nutno zapsat kombinaci znaků <?, kód jenž se píše je nutno mít za touto kombinací a až je dopsán opětovně uzavřít kombinací znaků ?>.

Za každý řádek kódu, který se napíše je zapotřebí napsat středník, není-li napsán kompilátor vyhodnotí chybu - *syntax error*.

Protože práce je z velké části zaměřená právě na tento jazyk je vhodné si popsat jeho základní prvky (2,7)

1.1.2.1 Základy

Mezi nutné základy pro správné použití jazyka patří komentáře. Vzhledem k tomu, že po několika tisících řádků kódu může samotný program připadat nezaujatému pozorovateli matoucí je tudíž vhodné vkládat průběžně vysvětlení ohledně funkčnosti jednotlivých částí kódu. Syntaxe pro komentáře jsou dvojího typu. Pokud má být komentář pouze na jednom řádku, využívá se dvou dopředných lomítek //, za které se napíše komentář.

```
602  
603 echo "</div>"; // sloupec  
604 echo "</div>"; // radek
```

Obrázek č. 1: Příklad jednořádkového komentáře

(Zdroj: vlastní zpracování)

Pro komentáře s více řádky se využívá dvojice znaků /* za kterými následuje komentář. Nesmí se zapomenout na uzavření komentáře reverzní dvojicí znaků */.

Samotný program by neměl smysl, pokud by neumožnil uživateli vypsat výsledky. To řeší příkaz `print(argument)` a `echo()`. Zatímco první varianta `print()` vrací hodnotu boolean, `echo` vrací void a dokáže vypsat několik řetězců. (2,10,11,12)

1.1.2.2 Proměnné

Nedílnou součástí každého programovacího jazyka patří proměnné. „Proměnná je pojmenované místo v paměti, které obsahuje data a dá se s ním manipulovat během vykonávání programu“. (2 str. 86)

Deklarace proměnné se provádí symbolem \$. Označení proměnné může být krom písmen či slov i číslo. Je důležité upozornit, že PHP není *case sensitive* jazyk, takže je nutné si dávat pozor na případná počáteční velká písmena. (2)

Stojí za zmínku, že PHP je dynamicky typovaný jazyk. Není nutné tedy na začátku programu deklarovat jednotlivé proměnné a jejich datový typ. (2)

1.1.2.3 Datové typy

„Datové typy je obecný název přiřazený nějaké množině dat, která sdílí nějakou společnou množinu charakteristik“ (2 str. 76). Obecně se rozdělují na skalární či složené.

Mezi skalární datové typy, které byly použity v rámci bakalářské práce patří zejména:

- Integer – pro celé číslo, bez desetinné čárky.
- Boolean – datový typ určený pouze pro dvě hodnoty – *true* či *false*.
- String – neboli řetězec by se dal definovat jako „posloupnost znaků se kterou se zachází jako s jednolitou skupinou.“. Posloupnost znaků je uzavřena v úvozovkách, jenž jednoznačně deklarují řetězec. Podrobněji je rozebrán v kapitole 1.1.2.5 Řetězce.

Mezi složené datové typy, které byly použity v rámci bakalářské práce patří zejména:

- Pole – často označována jako „indexovaná kolekce hodnoty dat“. Ke každému členu v poli se dá přistupovat jednotlivě pomocí jeho indexu, kdy index nemusí být pouze číslo ale i text.
- Objekt – hlavní prvek v objektově orientovaném programování. Definuje se na základě třídy, která určuje, jaké bude mít vlastnosti a k ní přiřazené metody. Vzhledem k tomu, že ohledně tematiky objektově orientovaného programování vznikly bezpočet publikací, nelze ji shrnout v pár větách a je podrobněji popsána v 1.1.2.8 Objektově orientované programování (2,13)

1.1.2.4 Podmínkové příkazy

V průběhu psaní kódu se vždy naskytne okamžik, kdy je potřeba rozvětvit kód podle určité podmínky. Syntax pro podmínkovou větev v PHP je:

```
if (vyraz) {  
  
    prováděná část kódu po splnění podmínky  
  
}
```

Pokud je splněna podmínka v závorce, program vykoná příslušný kód. Příkladem v bakalářské práci může být nastavení měny na Euro v případě, že informační systém identifikuje uživatele z ciziny (Obrázek 2).

```
102 if ($_SESSION["JAZYK"] == "en")  
103 {  
104     $_SESSION["MENA"]="Euro";  
105 }  
106
```

Obrázek č. 2: Ukázkový příklad použití příkazu if

(Zdroj: vlastní zpracování)

Výše zmíněný příklad se týkal použití pouze kladné větve – pokud je podmínka splněna, provede se kód a pokud není splněna, systém pokračuje dál ve své činnosti. Může ovšem nastat případ, kdy je zapotřebí aby systém vykonal příkazy v případě, že podmínka není splněna. K tomu účelu existuje i záporná větev podmínkové příkazu – *else*. Pro lepší demonstraci větve *else* poslouží následující příklad. V případě, že je nastaven výchozí jazyk systému v angličtině, nadpis jenž se zobrazí uživateli bude „ZEFIS - audit of information systems“, pokud nebude jazyk nastaven pro angličtinu systém změní nadpis na „ZEFIS - audit informačních systémů“ (Obrázek č. 3)

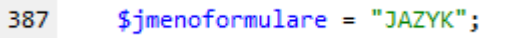
```
264 <?  
265 if ($_SESSION["JAZYK"] == "en")  
266 {  
267 ?>  
268     <title>ZEFIS - audit of information systems</title>  
269 <?  
270 }  
271 else  
272 {  
273 ?>  
274     <title>ZEFIS - audit informačních systémů</title>  
275 <?  
276 }  
277 ?>
```

Obrázek č. 3: Použití větve else

(Zdroj: vlastní zpracování)

1.1.2.5 Řetězce

Jak již bylo uvedeno v kapitole 1.1.2.3 jedná se o určitou posloupnost znaků. Na níže uvedeném obrázku lze vidět deklaraci řetězce (Obrázek č. 4).



```
387 $jmenoformulare = "JAZYK";
```

Obrázek č. 4: Deklarace řetězce

(Zdroj: vlastní zpracování)

PHP nabízí až přes 100 funkcí určené pro práci s řetězci. (2) Nejvíce využité byly v rámci práce především:

`strlen(retezec)` – funkce, jenž zjistí délku řetězce a vrátí počet znaků v řetězci (2)

`strFTIME()` – zajišťuje formátování dle místní časové zóny (2)

`substr(retezec, začátek, delka)` – vrátí část řetězce, která se nachází na pozici začátek a delka. V případě že nepovinný argument *delka* nebude uveden, funkce vrátí řetězec mezi začátkem až po konec řetězce (2)

`strcmp(retezec1, retezec2)` – slouží pro porovnání 2 řetězců, nebere v potaz velikost písmen. Často využíván pro kontrolu emailové adresy. (2)

`strtoupper(retezec)` - převede všechny znaky v řetězci na velká písmena (2)

1.1.2.6 Cykly

Pokud je potřeba opakovat určitou posloupnost příkazů, dokud se nesplní určitá podmínka přijde vhod mechanismus, jenž je v prakticky každém programovacím jazyku a tím je cyklus. V práci byly využity 2 základní typy cyklu `for` a `while`

`While` cyklus se provádí tak dlouho dokud není splněna určitá podmínka. (2) Syntax je následující

```
While(podmínka){
```

```
příkazy}
```

Druhým typem cyklu, který byl použit v práci je `for`. Na rozdíl od předcházející varianty je znám počet opakování. Opakování je svázáno s řídicí proměnnou, která při každém průchodu cyklem mění svoji hodnotu (2)

Na obrázku lze vidět názorný příklad použití for cyklu pro vypsání html příkazu. Na začátku cyklu je řídicí proměnné *i* přiřazena hodnota 0, poté co cyklus ověří zda není *i* menší než proměnná úroveň se zvedne hodnota *i* o jednotku a cyklus pokračuje dále.

```
608 for ($i=0;$i<$uroven;$i++)  
609     echo "<div class=col-md-1></div>";  
...
```

Obrázek č. 5: Ukázka for cyklu

(Zdroj: vlastní zpracování)

1.1.2.7 Funkce

Během vytváření práce bylo nutné řešit opakující se procesy. Bylo by velmi neuvážlivé kopírovat stále stejnou část kódu dokola, zvláště pokud by se změnila část samotného procesu, programátor či osoba zodpovědná za správu informačního systému by pak musela ručně procházet každou část kódu a tento proces nahrazovat novým. Tuto problematiku řeší programátorská konstrukce zvaná funkce. (2,14)

Pro vytvoření funkce v PHP se využívá následující syntaxe

```
function název_funkce (parametry) {  
  
    prováděná část kódu  
  
}
```

Je-li potřeba, aby funkce pracovala s určitými daty mimo funkci, je vhodné při deklaraci uvést i parametry, jenž budou využity v rámci funkce a za které pak budou dosazeny data při samotném volání funkce. (2)

Na následujícím obrázku (Obrázek č. 6) lze vidět deklarovanou funkci pro dokument. Vstupními argumenty jsou \$tabulka a \$nazev. Funkce provede několik operací s databází a co je podstatné, převeze parametry funkce a dosadí je v rámci kódu.

```

73 function dokument($tabulka,$nazev)
74 {
75     $this->id = "";
76     $this->aktivni = 0;
77     $this->vysledek = 0;
78     $this->vysledek_text = "";
79     $this->atributy="";
80     $this->klic == "";
81     $this->klic_hodnota == "";
82     $this->tabulka = $tabulka;
83     $this->nazev_tabulky = $nazev;
84     $this->strana = 1;
85     $this->stranka = 1;
86     $this->akce = "";
87     $this->subtabulky = "";
88     $this->subtabulka = "";
89     $this->lze_vyber = true;
90     $this->lze_editovat = true;
91     $this->lze_vkladat = true;
92     $this->lze_mazat = true;
93     $this->lze_zpet = true;
94     $this->smazat_tabulky = "";
95     $this->user_formular = "";
96
97     if ($tabulka <> "")
98         $this->nacteni_atributu($tabulka);
99
100 }
101

```

Obrázek č. 6: Příklad funkce s parametry

(Zdroj: vlastní zpracování)

Dalším příkladem je funkce poslicookie bez vstupních argumentů, která jak název napovídá smaže cookies (malé soubory ukládající se na straně uživatele kvůli procesu přihlašování na stránku (2)).

```

4 function smazcookie()
5 {
6
7     setcookie("id-jmeno", "", time() - 3600*24*5, '/');
8     setcookie("id-heslo", "", time() - 3600*24*5, '/');
9     setcookie("id-jazyk", "", time() - 3600*24*5, '/');
10    unset($_COOKIE["id-jmeno"]);
11    unset($_COOKIE["id-heslo"]);
12
13
14 }
--

```

Obrázek č. 7: Příklad funkce bez vstupních parametrů

(Zdroj: vlastní zpracování)

1.1.2.8 Objektově orientované programování

Jeden z hlavních programátorských konstruktů, na kterých je informační systém Zefis postaven je bezpochyby objektově orientované programování. To spočívá na třech základních principech – zapouzdření, dědění a polymorfismus.

Zapouzdření je mechanismus, kdy je kód skryt před uživatelem a ten přistupuje k potřebným částím přes rozhraní. Rozhraní se nazývají objekty, které se vytvoří na základě třídy (skrytého kódu). Třída nám definuje, jak má vypadat objekt a jeho atributy (Obrázek č.8). Metody jsou jinak nazvané funkce v OOP, které umožní uživateli manipulovat s daným objektem. (2)

```
2 class dotazniky
3 {
4     var $id;
5     var $idr;
6     var $ido;
7     var $idf;
8     var $idk;
9     var $ids;
10    var $idp;
11    var $idd;
12    var $typ;
13    var $stop;
14    var $povoleny;
15
16    var $rezim;
17    var $odpovezeno;
18    var $neodpovezeno;
19    var $lastidk;
20    var $lastido;
21
22    var $submenu;
23    var $vysledek;
24    var $vysledek_text;
25
26
27
```

Obrázek č. 8: Příklad třídy
(Zdroj: vlastní zpracování)

```

28
29  function dotazniky()
30  {
31      $this->id = 0;
32      $this->idr = 0;
33      $this->ido = 0;
34      $this->idk = 0;
35      $this->stop = false;
36      $this->povoleny = false;
37
38      $this->typ = "r";
39      $this->idd = 0;
40      $this->idf = 0;
41      $this->ids = 0;
42      $this->idp = 0;
43      $this->rezim = 0;
44      $this->odpovezeno = 0;
45      $this->neodpovezeno = 0;
46      $this->lastidk = 0;
47      $this->lastido = 0;
48
49      $this->submenu = 0;
50      $this->vysledek = 0;
51      $this->vysledek_text = "";
52  }
53
54

```

Obrázek č. 9: Metoda k předchozí třídě
(Zdroj: vlastní zpracování)

Dědičností se rozumí přenášení již existujících atributů tříd jiným třídám. Velmi zjednodušeně by se dal vysvětlit princip na příkladu - existuje-li třída *Obrazec* s atributy *rozmer* a *delka* a je vytvořena třída *Ctverec* jenž dědí ze třídy *Obrazec*, tak tato nově vytvořená třída bude mít též atributy *rozmer* a *delka* a není potřeba je opětovně definovat u této nově vzniklé třídy (2)

Polymorfismus – Velmi užitečná vlastnost OOP, umožňuje využívat rozhraní se stejným názvem ale s jinou implementací. (2)

1.1.3 MYSQL

Předchozí 2 technologie zajistili web, který dokáže být dynamický. Ovšem bez možnosti uložení dat zadaných do systému by projekt postrádal smysl. Tento problém řeší MySQL neboli My Structured Query Language. Jedná se o multiplatformní databázi, která funguje na principu relačního databázového modelu. (2)

Důvodů proč zvolit MYSQL je mnoho. Lze ho nasadit na prakticky jakýkoliv operační systém od nejznámějších širokou veřejností - Windows, Linux, Mac OS až po ty méně používanější jako je například Solaris. Pro některé uživatele může být příjemným

překvapením i rozsáhlá podpora různých jazyků. Velmi přínosným aspektem do budoucna je i silná podpora konfigurace a zabezpečení – lze nastavit vlastníka domény, velikost paměti přiřazená jednotlivým vláknům či ověření platného certifikátu SSL uživatele, jenž chce využívat naši databázi. Možností je velmi mnoho a s každou novou verzí lze očekávat jejich rozšíření a aktualizace. (2)

Komunikace mezi uživatelem a databází zajišťuje jazyk SQL. (2)

Pro přístupu k databázi byl využit volně dostupný nástroj phpMyAdmin, který využívám již mnoho let od dob mého studia na střední škole.

1.1.4 SQL

SQL neboli v překladu „Strukturovaný dotazovací jazyk“ je databázový jazyk pro operace s relační databází. (1)

„SQL je první a prozatím jediný standardizovaný databázový jazyk, který získal široké přijetí.“ (1, s. 76)

Většina informačních systémů je založených na SQL nebo aspoň jejich část. Hlavním cílem pro SQL je umožnění uživateli vytvářet tabulky, nahrávat a modifikovat data či provádět dotazy. (1)

„Standart SQL ISO má dvě hlavní součásti: Jazyk pro definici dat (Data Definitiv Language – pro definici struktury databáze a kontroly přístupu k datům

Jazyk pro manipulaci dat (Data Manipulation Language) pro vyvolání a aktualizaci dat “ (1 s. 77)

Samotná syntaxe se skládá z rezervovaných slov – např. SELECT, INSERT, které se musí napsat přesně tak jak jsou definovány jinak je dotaz vyhodnocen jako chybný a slov definovaných uživatelem – např. názvy tabulek či databází. Při psaní příkazů v SQL je dobré vzít v potaz, že většina příkazů není *case sensitive* tzn. Pokud se napíše příkaz WHERE či where, příkaz se provede v obou případech. (1)

1.1.4.1 Základní dotazy

SELECT – pro specifikaci sloupců, které mají být zpracovány (1)

FROM – pro specifikaci tabulek, z kterých se budou čerpat záznamy (1)

WHERE – pro určení podmínky na základě které mají být vyfiltrovány záznamy (1)

GROUP BY – sjednocuje řádky podle podmínky do skupiny. (1)

HAVING – filtruje skupiny. (1)

ORDER – řadí nalezené záznamy vzestupně – ORDER BY ASC či sestupně ORDER BY DESC (1)

1.1.4.2 Agregační funkce

Vrací jednoduchou hodnotu jako součet, či počet záznamů pro sloupce podle určitého kritéria. Lze použít pouze pro příkazy SELECT a HAVING. Nejčastěji používané jsou:

SUM – „vrací součet hodnot v zadaném sloupci“

COUNT - „vrací počet hodnot v zadaném sloupci“

AVG - „vrací průměr hodnot v zadaném sloupci“

MAX - „vrací největší hodnotu v zadaném sloupci“

MIN - „vrací nejmenší hodnotu v zadaném sloupci“ (1, str. 86)

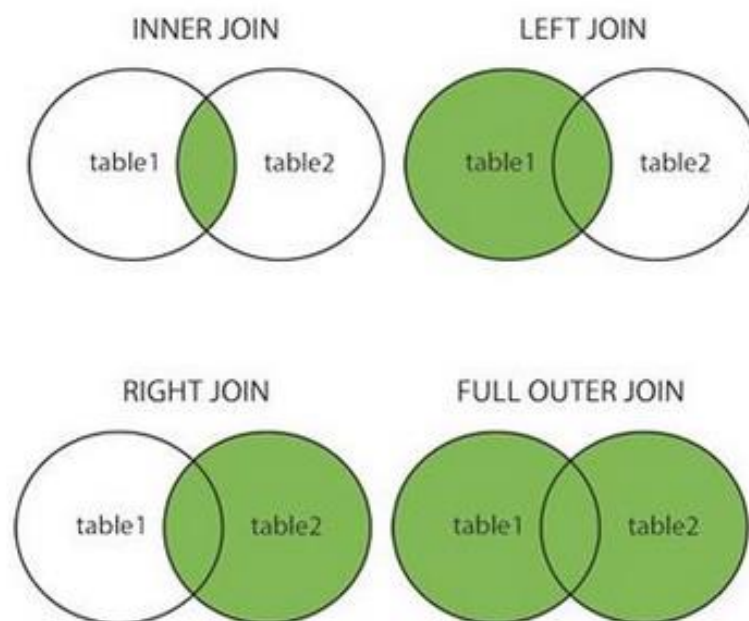
Pro funkční operabilitu databáze je nutné pracovat s více tabulkami současně, z tohoto důvodu se musí spojit. Druhy spojení tabulek máme 3 typy:

INNER JOIN – spojení, na základě kterého se zobrazí odpovídající záznamy jak z pravé tabulky tak i z levé tabulky. (1)

FULL JOIN – spojení, na základě kterého se zobrazí všechny záznamy jak z pravé tabulky tak i z levé tabulky. (1)

RIGHT JOIN – spojení, na základě kterého se zobrazí všechny záznamy z pravé tabulky a odpovídající záznamy z levé tabulky. (1)

LEFT JOIN – spojení, na základě kterého se zobrazí všechny záznamy z levé tabulky a odpovídající záznamy z pravé tabulky. (1)



Obrázek č. 10: Znázorněné průniky dat na základě spojení tabulek
(Zdroj: 15)

Pro operování s daty se využívají 3 základních příkazů pro vložení, aktualizaci či odstranění záznamů.

INSERT – vloží nové záznamy do tabulky (1)

UPDATE – aktualizuje záznamy v tabulce (1)

DELETE – odstraní záznamy z tabulky (1)

1.2 Základní pojmy

1.2.1 Data

V současné době se pojmy informace a data velmi často zaměňují, proto je nutné si tyto pojmy správně ujasnit. Pro jednoduché pochopení je příhodné si tyto pojmy vysvětlit na praktických příkladech. Pokud je napsáno slovo „zelená“, každému příjemci by bylo evidentní, že se jedná o barvu. Problém by nastal, pokud se napsalo 80-7197-217-7, tato čísla sama o sobě nic neznamenají a nikdo by rozhodně neuhádl, že se jedná o ISBN kód ke knize. Tedy, data nemají logický smysl a až poté co jim je přiřazen se z nich stávají informace. (1,2)

Obecně by se tedy dalo říct, že „data jsou surová (nezpracovaná) fakta, která mají určitou hodnotu pro organizaci“. (1, s. 36)

1.2.2 Informace

Jak již bylo zmíněno, proto aby mohla vzniknout informace je nutné disponovat daty, kterým bude posléze přiřazen význam. (1,2)

1.2.3 Databáze

Pod slovem databáze si lze představit organizované úložiště dat. K uloženým datům následuje i popis dat – tzv. metadata. Správně navrženou databázi charakterizuje nezávislost dat, tudíž pokud je přidán či odstraněn záznam v databázi, neměla by se narušit její struktura. Pro operování s databází je nutné použít Systém řízení databáze – v případě této práce byl využit MySQL popsáný v kapitole Použitých technologií. Existují různé typy databází každá vhodná pro jiný účel. V rámci práce byl využit typ relační databáze, která se více hodí pro účely projektu na rozdíl od např. objektově orientované databáze. Vhodným začátkem pro vytvoření relační databáze je sestavení relačního modelu. (1,2)

1.2.4 Relační model

Při návrhu relační databáze je nutné specifikovat co vše by měla obsahovat – jaká data a jejich návaznost. Při použití toho modelu jsou data rozmístěná do tabulek, kdy každá relace (tabulka) má svoje atributy (sloupce) a záznamy (řádky). Atributy mají svůj vlastní datový typ, podle kterého databázový systém rozpozná, jak s nimi má pracovat. Například pokud je zapotřebí ukládat celočíselné hodnoty je nutné nastavit datový typ jako smallinteger, integer či biginteger podle velikosti. Datových typů, které lze nastavit je velmi mnoho pro tuto práci však byly použity jen určité, jejichž příklady jsou uvedeny v tab.1. Nastavení správného datového typu je zásadní, protože systém potřebuje využívat správně funkce. Kupříkladu pokud by bylo uloženo číslo jako text, zamezilo by se použití funkce pro zjištění maximální hodnoty. Relace jsou mezi sebou vzájemně propojené. Aby mohlo spojení vzniknout je zapotřebí mít v každé tabulce primární klíč a v připojené tabulce cizí klíč. (1,2)

Tabulka č.1 Příklady použitých datových typů pro SQL
(Zdroj: vlastní zpracování dle: 16)

Název datového typu	Typ	Velikost
Int	Celočíselný typ	4 Bytů
Tinyint	Celočíselný typ	1 Byte
Char	Textový typ	Maximální velikost 8000 znaků
Date	Datum	3 Byty
Datetime	Kombinace data a času	8 Bytů
Text	Textový typ	Maximální velikost 65,536 znaků

1.2.5 Primární klíč

Atribut či množina atributů, jenž jednoznačně identifikuje záznam v tabulce. (1)

1.2.6 Cizí klíč

Jedná se o primární klíč cizí tabulky. (1)

1.2.7 Kardinalita vztahu

Spojení mezi tabulky nám omezuje kardinalita vztahu na následující vazby

1:1 – 1 záznamu v tabulce odpovídá 1 záznamu v cizí tabulce. Častým příkladem pro pochopení relace 1:1 je v literatuře uváděn následující příklad „jeden občanský průkaz je obdržén pouze jednou a právě jednou osobou“. (1)

1: N – 1 záznam z tabulky odpovídá více záznamům z druhé tabulky. (1)

M:N – Více záznamů v tabulce odpovídá více záznamům z druhé tabulky. V tomto případě nastává „dekompozice tabulky“ – kdy při tomto procesu dojde k vytvoření nové tabulky. V této nově vzniklé entitě jsou zapsané oba primární klíče z tabulek. Jedná se o jednu z nejobvyklejších relací a byla použita i v této práci. (1)

1.2.8 Entitně – relační model

Při návrhu databáze je důležité se zamyslet z pohledu všech koncových uživatelů, kteří budou s touto databází pracovat. Pakliže ji vytvoříme a zapomene na určitou skupinu, bude nutné kompletně změnit již hotovou práci a předělat ji. K vyhnutí této problematiky slouží proces nazvaný entitně – relační modelování, při kterém se „začíná určením důležitých dat (nazývaných entity) a relací mezi daty, které je třeba v modelu reprezentovat“. (1 s. 155)

V rámci bakalářské práce jsem vytvořil 2 ER modely dle Crow's Foo notace, v závislosti na splnění hlavních cílů práce. Oba jsou blíže popsány v kapitole Vlastní řešení

2 ANALÝZA SOUČASNÉHO STAVU

V rámci bakalářské práce mi bylo umožněno podílet se na vývoji informačního systému Zefis, provozovaného na www.zefis.cz. Informační systém Zefis existuje od roku 2010, je vytvářen Doc.Ing.Milošem Kochem, CSc. z Fakulty podnikatelské VUT v Brně. Hlavním cílem tohoto systému je dát firmám k dispozici nástroj, pomocí kterého si pomocí dotazníků mohou ověřit, zda používají informační systémy v souladu s best practices, a nedělají při jeho užívání chyby, které by ohrožovaly chod firmy a snižovaly efektivnost jeho využívání. Z pohledu autora pak jde o nástroj, pomocí kterého získává představu o stavu informačních systémů ve firmách a výsledky využívá ve své vědecké práci, společně s doktorandy a kolegy.

V původní verzi systému se Zefis zaměřoval pouze na informační systémy, a pomocí metody HOS8 ukazoval v osmi oblastech odchylky od ideálního stavu. Kromě těchto výsledků měli uživatelé možnost porovnat své odpovědi na jednotlivé otázky s odpověďmi ostatních firem, a získat tak představu, zda jsou v dané oblasti lepší, nebo horší než srovnatelné firmy.

Systém Zefis umožňuje firmám provést anonymně, při zachování plné důvěrnosti dat, základní diagnostiku chyb při užívání informačních systémů v oblastech:

- Technické vybavení
- Programové vybavení
- Pravidla
- Pracovníci
- Data
- Zákazníci
- Provoz

Technické vybavení

V této oblasti je zjišťováno, zda hardware, používané ve firmě, je adekvátní potřebám, a je dostačující pro efektivní práci uživatelů i informačních systémů.

Programové vybavení

Oblast zkoumá, zda programové vybavení vyhovuje potřebách uživatelů, je dostatečně aktuální, rychlé a spolehlivé.

Pravidla

Zde je zkoumáno, zda jsou definovaná pravidla, podle kterých firma funguje, ať již procesní pravidla, nebo ostatní, pokud není používáno procesní řízení, zda jsou tato pravidla známá pracovníkům, a zda jsou kontrolována a dodržována.

Pracovníci

V oblasti, zaměřené na pracovníci, se zkoumá jejich schopnost pracovat s informačními systémy, dodržování pravidel, školení, přístupová práva.

Data

Zde je zjišťována, zda jsou k dispozici všechna potřebná data, zda jsou chráněna a zálohována, případně klasifikovaná.

Zákazníci

V oblasti zákazníků se ověřuje dodržování ochrany dat zákazníků, GDPR, zda mají zákazníci dostupné potřebné funkce IS (pokud pro ně existuje)

Provoz

V oblasti provozu se zjišťuje, zda mají pracovníci k dispozici podporu při práci s IS, jak jsou s touto podporou spokojeni, zda dodržují pravidla, mají k dispozici všechny funkce IS, které potřebují ke své práci.

V nové verzi Zefis, realizované v létě 2020, při jejíž přípravě jsem řešil svoji bakalářskou práci, došlo k rozšíření pohledu na problematiku o procesy. Zefis nyní zkoumá čtyři základní okruhy:

- Firmu
- Informační systém
- Proces
- Užití informačního systému v procesu

K tomuto rozšíření došlo proto, že zkoumat efektivnost a efektivitu informačních systémů samo o sobě nelze, protože jejich přínosy, případně nedostatky, vznikají až užitím těchto

systémů v procesech firmy. Také se ukázalo, že zatímco v jednom procesu se systém může jevit jeho výborný, velmi užitečný, v jiném procesu může být jeho nasazení obtížné a nevyhovující. Proto Zefis nyní zkoumá informační systémy z pohledu procesů.

V systému jsou aktuálně využívány čtyři dotazníky. Tři z nich jsou dotazníky faktů, a jsou vyplňovány pouze jednou. Jedná se o dotazník firmy, systému a procesu. V těchto dotaznících je popisovaná skutečnost, například velikost firmy, kterou není třeba zjišťovat vícekrát. Druhý typ dotazníku, dotazník zkušeností, může být vyplněn více pracovníky, a popisuje subjektivní názor, například jak je pracovník spokojen s úrovní technické podpory, zda mu v systému nechybí některé funkce, potřebné pro jeho práci. Tento typ dotazníku je určen pro zkoumání, jak je užíván konkrétní informační systém v konkrétním procesu firmy.

2.1 Datový model

Následující část, popisující datový model, je zaměřena pouze na klíčové objekty, které jsou nezbytné pro pochopení funkčnosti systému, a jsou potřebné pro vlastní řešení práce.

2.1.1 Tabulka Z_FIRMY


Tato tabulka obsahuje základní data o firmě. Cizí klíč *idu* spojuje tuto tabulku s uživatelem systému, který ji vytvořil. Přístup od – do určuje období, ve kterém je možné s firmou pracovat, standartně tři měsíce v neplacené verzi portálu.

Z_Firmy	
 PK id	int(11)
nazev	text
datum	date
poznamka	text
idu	int(11)
pristupod	date
pristupdo	date
zaplaceno	tinyint(4)
zaplaceno1	tinyint(4)
zeme	int(11)

Obrázek č. 11 Tabulka Z_FIRMY
(Zdroj: vlastní zpracování)

2.1.2 Tabulka Z_SYSTEMY


Tabulka definuje systém. Obsahuje cizí klíče *idu* na tabulku uživatelů a *idf* na tabulku firmy, ve které je používán.

Z_Systemy	
 PK id	int(11)
idf	int(11)
idu	int(11)
nazev	text

Obrázek č. 12 Tabulka Z_SYSTEMY
(Zdroj: vlastní zpracování)

2.1.3 Tabulka Z_PROCESY

Tabulka obsahuje základní data o procesech firmy. Stejně jako tabulky systémů obsahuje cizí klíče na tabulky firem a uživatelů.

Z_Systemy	
 PK id	int(11)
nazev	text
idu	int(11)
idf	int(11)

Obrázek č. 13 Tabulka Z_SYSTEMY
(Zdroj: vlastní zpracování)

2.1.4 Tabulka Z_AUDITY

V systém jsou čtyři audity, založené na čtyřech typech dotazníků. Audit firmy, systému, procesu a užití systému v procesu. Tabulka audity definuje audit, a používá cizí klíče na tabulky:

Idf – id firmy


Idu – id uživatele

Idm – id dotazníku

Ids – id systému

Idp – id procesu

Takto nazvané cizí klíče jsou pak použity ve všech ostatních tabulkách.

Z_Audity	
 PK id	int(11)
idf	int(11)
idu	int(11)
idm	int(11)
kod	text
nazev	text
nazev_en	text
datumod	date
datumdo	date
poznamka	text
stav	int(11)
pristupdo	date
pristupod	date
idp	int(11)
ids	int(11)
typ	char(1)
vypocet	tinyint(4)

Obrázek č. 14 Tabulka Z_AUDITY
(Zdroj: vlastní zpracování)

K auditu je připojen buď jeden respondent, který vyplnil dotazník – audit faktů (firma, systém, proces), nebo více respondentů u auditu užití (užití systému v procesu).

2.1.5 Tabulka Z_RESPONDENTI

Tabulka respondentů představuje jeden vyplněný dotazník. Neobsahuje otázky, ani odpovědi, pouze definuje typ dotazníku, a jaké firmy, systému a procesu se týká. Obsahuje cizí klíče na tabulky:

Idd – id dotazníku


Ida – id auditu

Idf – id firmy

Ids – id systému

Idp – id procesu

Podle typu dotazníku jsou některé cizí klíče nepoužité – u dotazníku firmy nejsou vazby na systém a proces, u dotazníku systému na proces, u dotazníku procesu na systém. U dotazníku užití systému v procesu jsou vyplněny všechny cizí klíče.

Z_Respondenti	
 id	int(11)
kod	char(10)
idd	int(11)
ida	int(11)
stav	int(11)
zahajen	datetime
odevzdan	datetime
idf	int(11)
typ	char(1)
profese	char(1)
autor	text
popis	text
ids	int(11)
idp	int(11)


Obrázek č. 15 Tabulka Z_RESPONDENTI
(Zdroj: vlastní zpracování)

2.1.6 Tabulka Z_ODPOVEDI

Tato tabulka obsahuje odpovědi respondenta na otázky dotazníku. Odpověď se ukládá jako jeden znak (typicky volba a, b), a tabulka obsahuje cizí klíče na:

Idr – id respondenta z tabulky respondentů

Ido – id otázky z tabulky otázek

Z_Odpovedi	
 PK id	int(11)
idr	int(11)
ido	int(11)
odpoved	char(1)

Obrázek č. 16 Tabulka Z_ODPOVEDI
(Zdroj: vlastní zpracování)

2.1.7 Tabulka Z_OTAZKY

Tabulka obsahuje všechny otázky dotazníků. Každá otázka má svoje id, na které jsou navázány odpovědi na tuto otázku v tabulce Z_ODPOVEDI.

Z_Otazky	
 PK id	int(11)
otazka	text
typ	tinyint(2)
popis	text
otazka_en	text
popis_en	text
skupina	int(11)
oblast	char(1)
povinna	tinyint(5)
bezpecnost	tinyint(4)
poradi	smallint(6)
ukazat	tinyint(4)
rokod	int(11)

Obrázek č. 17 Tabulka Z_OTAZKY
(Zdroj: vlastní zpracování)

2.1.8 Tabulka Z_MOZNOSTI


Tato tabulka definuje možné odpovědi na otázku. Obsahuje cizí klíč *ido* – id otázky.

V atributu *klic* je uveden kód odpovědi, který se pak ukládá do tabulky Z_ODPOVEDI. Atribut *volba*, resp. *volba_en* popisuje tuto volbu v dotazníku. Systém obsahuje českou a anglickou verzi, proto tyto texty jsou uloženy ve dvou attributech.

Z_Moznosti	
 PK id	int(11)
ido	int(11)
klic	char(1)
volba	text
volba_en	text
typ	int(11)

Obrázek č. 18 Tabulka Z_MOZNOSTI
(Zdroj: vlastní zpracování)

Ukázka otázky, která má dvě možné odpovědi – a,b v dotazníku.

id	 Klíč	Volba	Volba EN
1027	a	Ano	Yes
1028	b	Ne	No

Obrázek č. 19 Příklad otázky s dvěma možnostmi
(Zdroj: vlastní zpracování)

2.2 Audity

V následující části je popsáno, jak probíhá sběr dat – audity v systému ZEFIS.

Nový audit

Q

Audit

Výpočet

Stav

Id	Audit	Firma	Výpočet	Stav	Proces	Systém
2074	Audit užití	3812	Ano	Dokončený	Administrativa	apollo
2069	Audit procesu	3812	Ano	Dokončený	Administrativa	
2061	Audit systému	3812	Ano	Dokončený		apollo
2060	Audit firmy	3812	Ne	Rozpracovaný		

Obrázek č. 20 Ukázka auditů v systému
(Zdroj: vlastní zpracování)

Průzkum je založen na vyplnění čtyř dotazníků u čtyř auditů. Po rozkliknutí auditu se zobrazí dotazník, případně více dotazníků u auditu užití.

Audit 2060 | Informace o firmě

Audit byl dokončen 08.01.2021. Můžete si u něj spočítat a zobrazit výsledky v menu **Mé výsledky**. V tomto auditu máte **1 dotazník**, z toho **1 odevzdaných**. Tento audit má zaplacený plný přístup do **20.09.2021**. Po tomto datu již nelze přidávat další dotazníky, ale pouze prohlížet výsledky.

Dotazníky auditu

Pro práci s dotazníkem vyberte dotazník v tabulce a **klepnutím** ho otevřete

Id	Stav	Firma	Autor	Proces	Systém	Shoda	Chybí otázek
14132	Dokončený	3812	koch@zefis.cz			vysoká	0

Obrázek č. 21 Dotazník k auditu
(Zdroj: vlastní zpracování)

2.3 Dotazník

Při vyplňování dotazníků jsou pokládány otázky podle scénáře dotazníku. Scénář dotazníku definuje, které otázky a v jakém pořadí se budou zobrazovat, a zahrnuje i logické podmínky. Pokud je logická podmínka splněna, například respondent odpověděl na některou otázku určitým způsobem, jsou položeny některé otázky, který se jinak v dotazníku nezobrazí. Na obrázku je uvedena ukázka části scénáře dotazníku. V tomto scénáři je položena otázka 46 pouze v případě, že na otázku 189 odpověděl respondent volbou a.

Krok:26	Otázka:159			Může ve vaší organizaci někdo, kdo není zaměstnanec, připojit své zařízení (notebook, mobil) do vaší firemní počítačové sítě?
Krok:27	Otázka:189			Ukládají si uživatelé na disky svých počítačů ve firmě data
Krok:28	Otázka:189	IF	a	Ukládají si uživatelé na disky svých počítačů ve firmě data Ano
Krok:29		Otázka:46		Probíhá ve Vaší organizaci pravidelné zálohování dat uložených na lokálních počítačích pracovníků?
Krok:30	Otázka:189	ENDIF		
Krok:31	Otázka:93			Je ve vaší organizaci vypracována metodika zálohování dat?

Obrázek č. 22 Otázka 46

(Zdroj: vlastní zpracování)

U každé otázky v dotazníku respondent vybere jednu, nebo více odpovědí, podle typu otázky, a tato odpověď se uloží do tabulky Z_ODPOVEDI. Respondent může kdykoli vyplňování dotazníku přerušit, a pokračovat v něm později. V takovém případě si může zvolit pro zobrazení dotazníku možnost nezobrazovat všechny otázky, ale pouze nezodpovězené.

V systému se používají typy otázek:

- Otázka s jednou možnou odpovědí
- Otázka s více možnými odpověďmi
- Volná odpověď

Volné odpovědi (text) nejsou ukládány do tabulky Z_ODPOVEDI, ale do tabulky Z_ODPOVEDIT, s analogickou strukturou, ale s polem pro odpověď typu text a ne char.

Mé audity

Úvod

Mé firmy

Mé systémy

Mé procesy

Mé audity

Mé výsledky

Mé objednávky

Ukončit práci s dotazníkem

Odpovězte prosím na všechny otázky. Po poslední otázce se dotazník odešle a ukončí

Smazat odpověď

Otázka

Jak budeme chápat firmu, pro kterou děláte interní audit

Firmu tvoří dílčí části s určitou formou samostatnosti, které mají své jméno (ne útvary nebo oddělení). Musíme se rozhodnout, kterou část budeme v auditech hodnotit. Zda budete popisovat **jednu z těchto částí**, kde pracujete. (**Příklad:** firma bude Škoda, ale ne koncern Volkswagen), nebo budete popisovat celek (**Příklad:** firma bude Volkswagen, ale ne Škoda).

- ☐ Firma, kterou budu popisovat, není součástí žádného vyššího celku
- ☒ Firmu tvoří dílčí části, já budu popisovat **jednu z těchto částí**
- ☐ Firmu tvoří dílčí část, já budu popisovat **celou firmu**, ne její část

Odeslat

Obrázek č. 23 Příklad otázky v systému
(Zdroj: vlastní zpracování)

Mé audity

Úvod

Mé firmy

Mé systémy

Mé procesy

Mé audity

Mé výsledky

Mé objednávky

Ukončit práci s dotazníkem

Odpovězte prosím na všechny otázky. Po poslední otázce se dotazník odešle a ukončí

Smazat odpověď

Předchozí otázka

55%

Otázka

Pro které nosiče dat máte definována opatření pro jejich likvidaci? (metodiky, pokyny, vnitřní předpis, instrukci apod.) Můžete vybrat více odpovědí.

- ☒ listiny a jiné analogové dokumenty, např. knihy, kartotéky, fotografie
- ☐ pevné disky
- ☐ magnetické nosiče dat (diskety, magnetofonové pásky, audiokazety a videokazety, karty s magnetickým proužkem)
- ☐ optické nosiče dat (CD, DVD)
- ☐ ostatní elektronické nosiče dat (USB disky, čipové karty, paměťové karty apod.)
- ☐ kompletní počítače kategorie servery
- ☐ kompletní počítače kategorie PC
- ☐ kompletní počítače kategorie notebook
- ☐ tablety
- ☐ mobilní telefony
- ☐ kompletní skenery a kopírky
- ☐ síťová zařízení (router, switch, modem a podobně)
- ☐ data nacházející se u externích dodavatelů (poskytovatelů služeb), např. cloudových
- ☐ jiné nosiče dat, výše neuvedené

Odeslat

Obrázek č. 24 Příklad otázky v systému s vyšším počtem možných odpovědí
(Zdroj: vlastní zpracování)

2.4 Výsledky

Po vyplnění dotazníků a dokončení všech auditů dostává uživatel výsledky odpovědí na jednotlivé otázky, porovnané s odpověďmi ostatních firem. Aktuálně je v systému přibližně 3800 firem a 11800 respondentů (u auditu užití odpovídá na otázky více pracovníků firmy).

Výsledky jsou získány pomocí SQL dotazů z tabulky Z_ODPOVEDI, jako count (počet výskytů) pro jednotlivé varianty odpovědí

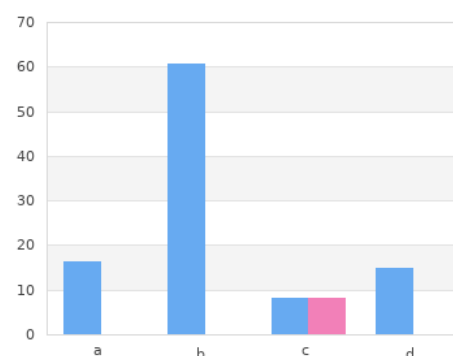
Tyto výsledky může uživatel využít pro představu, jak vypadá situace celkově, a získat tak představu, zda stav v jeho firmě odpovídá obvyklému stavu i jinde, nebo je jeho firma odlišná od ostatních. Z otázek nicméně nevyplývá, jaké by mělo být správné řešení. Proto byl systém ZEFIS rozšířen o zjištěné nedostatky, jež jsou řešeny v bakalářské práci

Otázka

Uvedte, které pracovní pozice ve vaší firmě provádí činnosti/ úkony související s bezpečností ICT (analýza bezpečnostních rizik, řešení bezpečnostních problémů, školení zaměstnanců týkajících se bezpečnosti ICT apod.)

Volba	Odpověď	Můj audit	Ostatní firmy
a	Tuto oblast v naší firmě nikdo nezajišťuje	0%	16.4%
b	Zaměstnanci firmy	0%	60.7%
c	Zaměstnanci mateřské firmy či jiných příbuzných firem v rámci skupiny podniků	100%	8.2%
d	Externí dodavatelé	0%	14.8%

Můžete si nechat srovnat vaše odpovědi s ostatními firmami. buď se všemi, nebo s firmami srovnatelnými s vaší.



Červenou barvou jsou znázorněny vaše odpovědi.

Obrázek č. 25 Výsledek odpovědí na otázku
(Zdroj: vlastní zpracování)

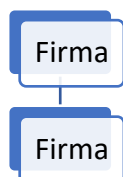
3 VLASTNÍ ŘEŠENÍ

V této části bude rozebrán způsob, jakým byly vyřešeny vytyčené cíle bakalářské práce.

3.1 Vícekriteriální vyhledávání

První z cílů práce je umožnit vícekriteriální vyhledávání v otázkách dotazníků, v závislosti na jiných odpovědích. Po analýze všech čtyř dotazníků se ukazuje, že lze používat tyto kombinace otázek.

Pohled FIRMA

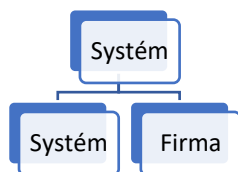


Obrázek č. 26 Pohled z hlediska firmy

(Zdroj: vlastní zpracování)

U otázek, týkajících se firmy, je možné používat pouze otázky firmy, neboť ani systémy, ani procesy nemají žádný vztah k faktům o firmě jako takové.

Pohled SYSTÉM

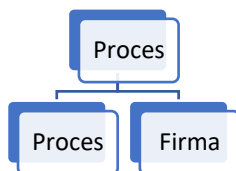


Obrázek č. 27 Pohled z hlediska systému

(Zdroj: vlastní zpracování)

U otázek na systém je možné současně vyhledávat také podle firmy, protože informační systém má relevantní vztah na firmu, ale fakt o systému, například stáří systému, nemá vztah na některý fakt o procesu.

Pohled PROCES



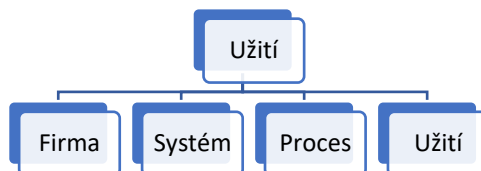
Obrázek č. 28 Pohled z hlediska procesu

(Zdroj: vlastní zpracování)

U otázek na proces je možné vyhledávat také související otázky podle firmy.

Pohled UŽITÍ

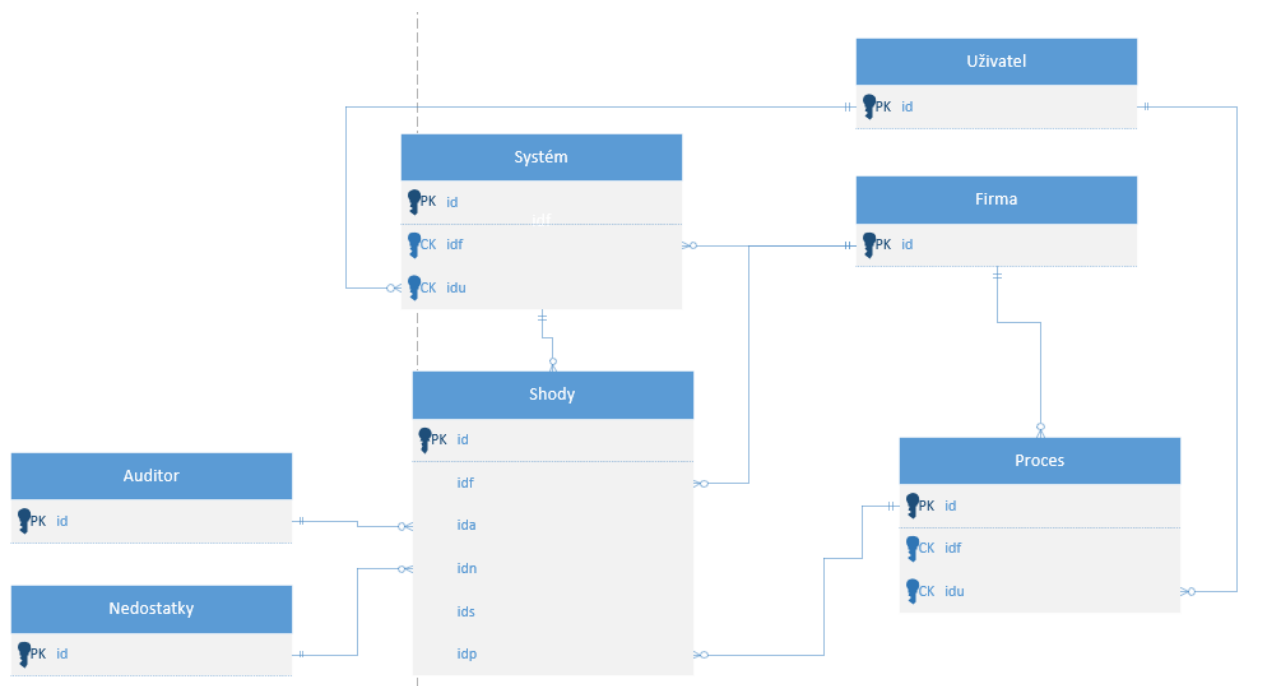
Při vyhledávání v otázkách dotazníku užití informačního systému v procesu firmy jsou relevantní související otázky také na firmu, systém a proces. Jde o nejkomplexnější pohled.



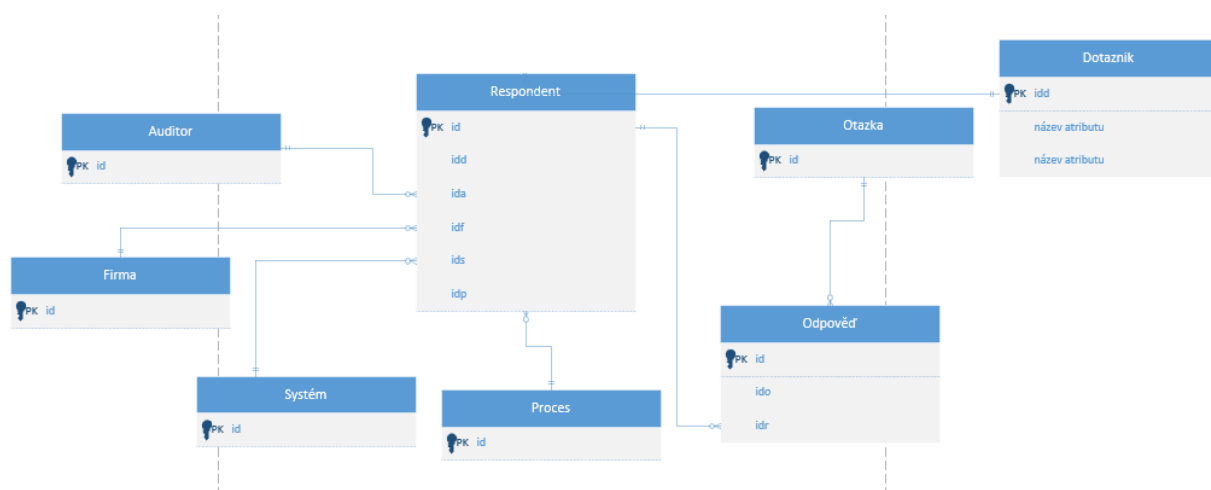
Obrázek č. 29 Pohled z hlediska užití
(Zdroj: vlastní zpracování)

3.2 Návrh Entitně-relačního modelu

Pro každý z cílů projektu byl navržen vlastní ER model sestávající z hlavních tabulek, které se k jednotlivým cílům práce vztahují. Na obrázku 29 je výsek z ER modelu, kde byly zavedeny nové tabulky k zjišťování nedostatků ve firmách, na obrázku 30 pak výsek ER diagramu, který ukazuje vztah upraveného ukládání odpovědí na otázky, které je nutné pro vícekritériální vyhodnocování.



Obrázek č. 30 ER model pro zjišťování nedostatků
(Zdroj: vlastní zpracování)



Obrázek č. 31 ER model úpravy struktury odpovědí na otázky
(Zdroj: vlastní zpracování)

3.3 Úprava datových struktur

Stávající způsob uložení dat a vyhledávání v nich neumožňuje vyhledávání podle odpovědí na více otázek současně.

Odpovědi na otázky jsou uloženy v tabulce Z_ODPOVEDI, kde *ido* je id otázky a *odpoved* je kód odpovědi, obvykle a, b, c ...

Z_Odpovedi	
PK id	int(11)
idr	int(11)
ido	int(11)
odpoved	char(1)

Obrázek č. 32 Tabulka Z_ODPOVEDI
(Zdroj: vlastní zpracování)

Pokud se zjišťuje, kolik respondentů odpovědělo na otázku například 11 volbou a, pak lze jejich počet zjistit jednoduchým sql dotazem

SELECT count(id) as pocet FROM Z_ODPOVEDI WHERE ido = 11 and odpověď = 'a'

Problémem je, že mezi otázkami a odpověďmi není žádný vztah, takže není možné pomocí jednoho sql dotazu omezit výběr odpovědí v kombinaci s jinou otázkou. Proto byly upraveny datové struktury tak, aby byl možný současný výběr podle více odpovědí na různé otázky pomocí rychlých a jednoduchých sql dotazů.

Podstatou úpravy je doplnění odpovědí přímo k vlastním objektům, tedy k firmě, systému i procesu. Do každého z těchto objektů byly doplněny atributy Pi, kde i je id otázky, aby bylo možno jednoznačně zjistit odpověď na otázku u daného objektu.

Toto řešení je možné, protože u tří hlavních objektů, firmy, systému i procesu se dotazníky týkají faktů o těchto objektech, a ke každému objektu existuje pouze jeden dotazník, a tedy i jedna odpověď na danou otázku.

3.3.1 Otázka s jednou možnou odpovědí

Při vyhledávání odpovědí firem je často používané kritérium velikost firmy, je potřeba zjistit, jak se liší odpovědi na libovolnou otázku podle velikosti firmy. Velikost firmy je dána v systému otázkou s id 101. Na tuto otázku je možná pouze jedna odpověď z množiny:

id	Skupina	Oblast	Pořadí	typ	Ukázat	Bezpečnost	Povinná	Otázka
101	Údaje firmy	Pravidla	5	Jedna odpověď	Ne	Ne	Ano	Kolik přibližně zaměstnanců má Vaše firma?

Obrázek č. 33 Příklad otázky s jednou možnou odpovědí

(Zdroj: vlastní zpracování)

↓ Klíč	Volba
a	méně než 10
b	10-49
c	50-99
d	100-199
e	200-499
f	500-999
g	více než 1000

Obrázek č. 34 Výčet možných odpovědí na otázku
(Zdroj: vlastní zpracování)

Do objektu Z_FIRMY tedy doplníme parametr

☐ 13 p101 char(1) cp1250_general_ci

Obrázek č. 35 Doplněný parametr
(Zdroj: vlastní zpracování)

Do tabulky je pak uložena hodnota odpovědi z dotazníků, tedy např. a

zeme	p178	p179	p101	p102	p124	p133	p174	p134
1			g	d	a	g		e
1			g	d	c	g		e
1			g	d	a	g		e

Obrázek č. 36 Uložené odpovědi z dotazníků
(Zdroj: vlastní zpracování)

3.3.2 Otázky s více odpověďmi

V systému jsou používány i otázky, kde je možné více odpovědí (voleb). Příkladem může být otázka 242 na zjišťování likvidace nosiče s využitím externího dodavatele.

242	Údaje firmy	Data	0	Více odpovědí	Ne	Ano	Ne	Uvedte prosím, pro které uvedené nosiče/ možnosti, provádíte tuto likvidaci s využitím externího dodavatele.
-----	-------------	------	---	---------------	----	-----	----	---

Obrázek č. 37 Příklad otázky s více možnými odpověďmi

(Zdroj: vlastní zpracování)

U této otázky může respondent zaškrtnout více odpovědí:

Ukázka části možných odpovědí:

⌵ Klíč Volba	
a	listiny a jiné analogové dokumenty, např. knihy, kartotéky, fotografie
b	pevné disky
c	magnetické nosiče dat (diskety, magnetofonové pásky, audiokazety a videokazety, karty s magnetickým proužkem)
d	optické nosiče dat (CD, DVD)
e	ostatní elektronické nosiče dat (USB disky, čipové karty, paměťové karty apod.)

Obrázek č. 38 Výčet možných odpovědí na otázku

(Zdroj: vlastní zpracování)

Tato skutečnost byla v systému zachycena opět pomocí atributu Pi, ale typu varchar, který umožňuje vložit řetězec znaků.

<input type="checkbox"/>	74 p242	varchar(30) cp1250_general_ci
--------------------------	---------	-------------------------------

Obrázek č. 39 Doplněný atribut

(Zdroj: vlastní zpracování)

p259	p260	p261	p242
b	b	f	dghjl
b	d	e	bfg
d	d	g	fhjkl

Obrázek č. 40 Uložené odpovědi z dotazníků

(Zdroj: vlastní zpracování)

Do systému byla doprogramována jednoduchá volba, která umožní při zavedení nové otázky do systému doplnit atribut do příslušné tabulky objektů.

Přidat atribut CHAR	Přidat atribut VARCHAR
----------------------------	-------------------------------

Obrázek č. 41 Doprogramovaná tlačítka na přidání atributu

(Zdroj: vlastní zpracování)

```

if (isset($_REQUEST["vytvorattr"]))
{
    $ot = $_REQUEST["prootazku"];
    $data = vetal("Z_OTAZKY",$ot);
    $skup = $data->skupina;
    $sql = "";
    $otj = "p";
    $otj .= $ot;
    if ($skup == 2) $sql = "ALTER TABLE `Z_FIRMY` ADD $otj CHAR(1) NOT NULL ;";
    if ($skup == 3) $sql = "ALTER TABLE `Z_SYSTEMY` ADD $otj CHAR(1) NOT NULL ;";
    if ($skup == 4) $sql = "ALTER TABLE `Z_PROCESY` ADD $otj CHAR(1) NOT NULL ;";
    if ($sql <> "") mysql_query($sql,$db);
    echo $sql;
}

if (isset($_REQUEST["vytvorattr1"]))
{
    $ot = $_REQUEST["prootazku"];
    $data = vetal("Z_OTAZKY",$ot);
    $skup = $data->skupina;
    $sql = "";
    $otj = "p";
    $otj .= $ot;
    if ($skup == 2) $sql = "ALTER TABLE `Z_FIRMY` ADD $otj VARCHAR(30) NOT NULL ;";
    if ($skup == 3) $sql = "ALTER TABLE `Z_SYSTEMY` ADD $otj VARCHAR(30) NOT NULL ;";
    if ($skup == 4) $sql = "ALTER TABLE `Z_PROCESY` ADD $otj VARCHAR(30) NOT NULL ;";
    if ($sql <> "") mysql_query($sql,$db);
    echo $sql;
}

```

Obrázek č. 42 Upravený kód v modulu kontroly dotazníku

(Zdroj: vlastní zpracování)

Do modulu kontroly dotazníku byla přidána část kódu, která po odeslání dotazníků na závěr přenesou odpovědi k dané firmě, systému či procesu do příslušných atributů.

3.4 Způsob vyhledávání

Způsobem, uvedeným v předchozí části, byly upraveny objekty Z_FIRMY, Z_SYSTEMY a Z_PROCESY, podle kterých půjde nyní vyhledávat s využitím více kritérií současně. Tento způsob nelze použít u objektu Užití systému v procesu, protože tam na otázky odpovídá více respondentů a k jedné otázce nelze přiřadit jednu odpověď. Vícekritériální vyhledávání u tabulek faktů nyní neprobíhá s využitím tabulky Z_ODPOVEDI, kam se ukládají odpovědi z dotazníků, ale z tabulek firmy, systému a procesu.

Příklad

Je potřeba z tabulky Z_FIRMY zjistit počet firem, které jsou menší než 10 zaměstnanců, a současně jsou z oblasti vzdělávání, nebo finančnictví.

Velikost firmy je atribut p101, méně než 10 zaměstnanců volba a

Oblast firmy je atribut p102, vzdělávání je volba d a finančnictví volba f

Sql příkaz nyní bude vypadat takto:

```
SELECT count(id) as pocet FROM Z_FIRMY where p101 = 'a' and (p102 = 'd' or p102 = 'f')
```

3.4.1 Funkce systému

V systému bylo doprogramováno vyhledávání v otázkách v závislosti na nastavených volbách. Do systému byla doplněna možnost přidat libovolnou prohlíženou otázku, včetně voleb, které uživatele zajímají, do tabulky navolených podmínek výběru.

Detail otázky

Nastavení výběru otázky

Kolik přibližně zaměstnanců má Vaše firma?

☒ méně než 10

☐ 10-49

☐ 50-99

☐ 100-199

☐ 200-499

☐ 500-999

☐ více než 1000

☒ Otázka je aktivní a bude použita ve výběru

Uložit nastavení

Odebrat otázku z výběru

Otázka bude použita ve všech skupinách

Obrázek č. 43 Příklad otázky

(Zdroj: vlastní zpracování)

Další výběrová kritéria

Přidej aktuální otázku

Nepoužito

Aktivní

101 - Kolik přibližně zaměstnanců má Vaše firma?
méně než 10

Obrázek č. 44 Ukázka informace o stavu navolené podmínky

(Zdroj: vlastní zpracování)

Na obrázku 43 je pak ukázáno, jak se toto nastavené kritérium zobrazuje v jiných otázkách. Pokud je kritérium zapnuté, je indikován stav Aktivní, a pokud kritérium nelze u dané otázky použít, je indikován stav Nepoužito. To je případ situace, kdy například kritérium zjišťuje, zda jde o některý typ procesu, a mělo by být použito v otázce, týkající se informačního systému, kde není žádná logická souvislost.

Při zobrazení jiné otázky, pokud lze nastavené kritérium logicky použít, se zobrazují výsledky všech firem, a jako výběr těch, které splňují nastavené kritérium

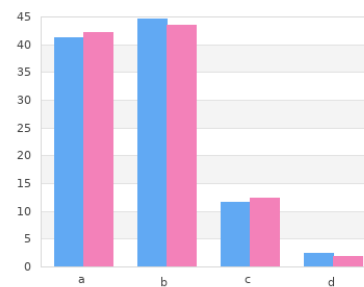
Použito

Aktivní

101 - Kolik přibližně zaměstnanců má Vaše firma?
méně než 10

Umožňuje vaše technika (počítače, servery) dobrou rychlosti a použitelnost vašich informačních systémů?

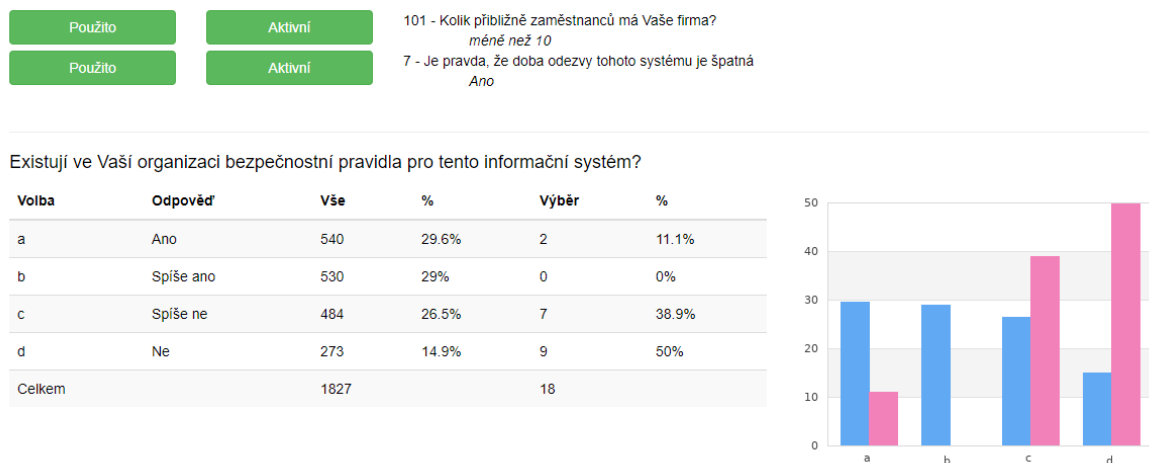
Volba	Odpověď	Vše	%	Výběr	%
a	Ano	774	41.2%	133	42.2%
b	Spiše ano	840	44.7%	137	43.5%
c	Spiše ne	220	11.7%	39	12.4%
d	Ne	45	2.4%	6	1.9%
Celkem		1879		315	



Obrázek č. 45 Zobrazené výsledky při použití jednoho kritéria

(Zdroj: vlastní zpracování)

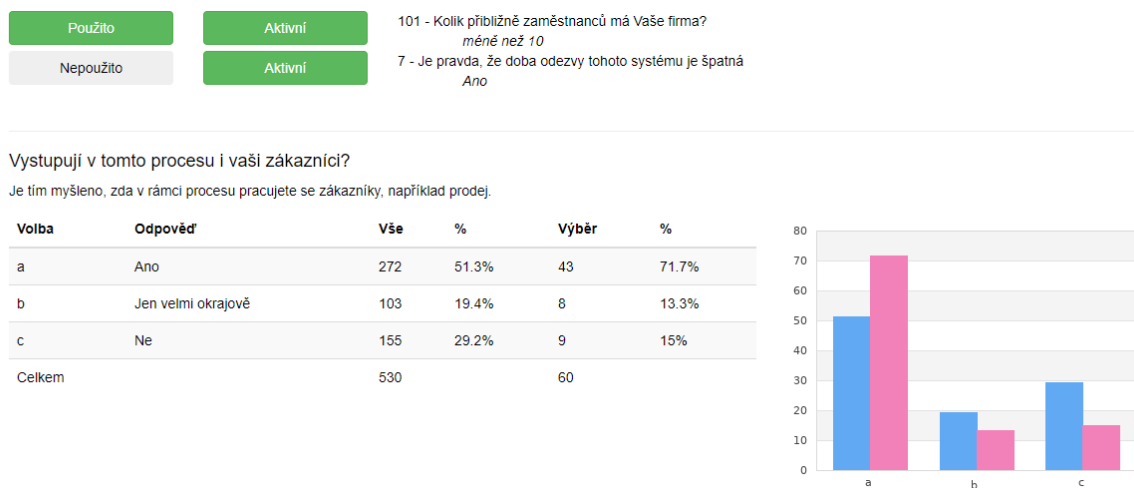
Pokud je přidáno další kritérium, u systému, pak výběr vypadá následovně. Otázka se týká systému, takže u ní lze použít obě kritéria – z firmy i systému.



Obrázek č. 46 Zobrazené výsledky při použití více kritérií

Zdroj: vlastní zpracování)

Pokud by se ale zjišťovali odpovědi na otázku, týkající se procesu, při takto nastavených výběrových kritériích, systém parametry, které se v otázkách faktů procesu zjišťovat nedají (u procesu se dají parametrizovat pouze otázky procesu a firmy, systémů může být v procesu více), automaticky vypne a označí je jako nepoužité.



Obrázek č. 47 Zobrazené výsledky při použití neshodných kritérií

(Zdroj: vlastní zpracování)

Naprogramováním této funkce byl splněn dílčí cíl práce, umožnit vyhledávání odpovědí v dotaznících podle více kritérií.

3.5 Identifikace nedostatků

Hlavním cílem práce je rozšířit systém Zefis o možnost identifikovat nedostatky, které existují ve firmách, systémech a procesech. Nedostatek, nebo lépe podle terminologie auditu neshoda se stavem, doporučeným podle osvědčených postupů, případně normám (best practices), je možné v systému diagnostikovat pouze podle odpovědí na otázky dotazníků, s využitím logických operátorů a současně a nebo.

Neshody v systému bude definovat expert, přičemž je nutné, aby formulace neshod nebyla závislá na programu a nutnosti kódovat podmínky neshody přímo v něm. Jinak by nebylo možné snadno doplňovat a definovat nové neshody, případně upravovat jejich definici.

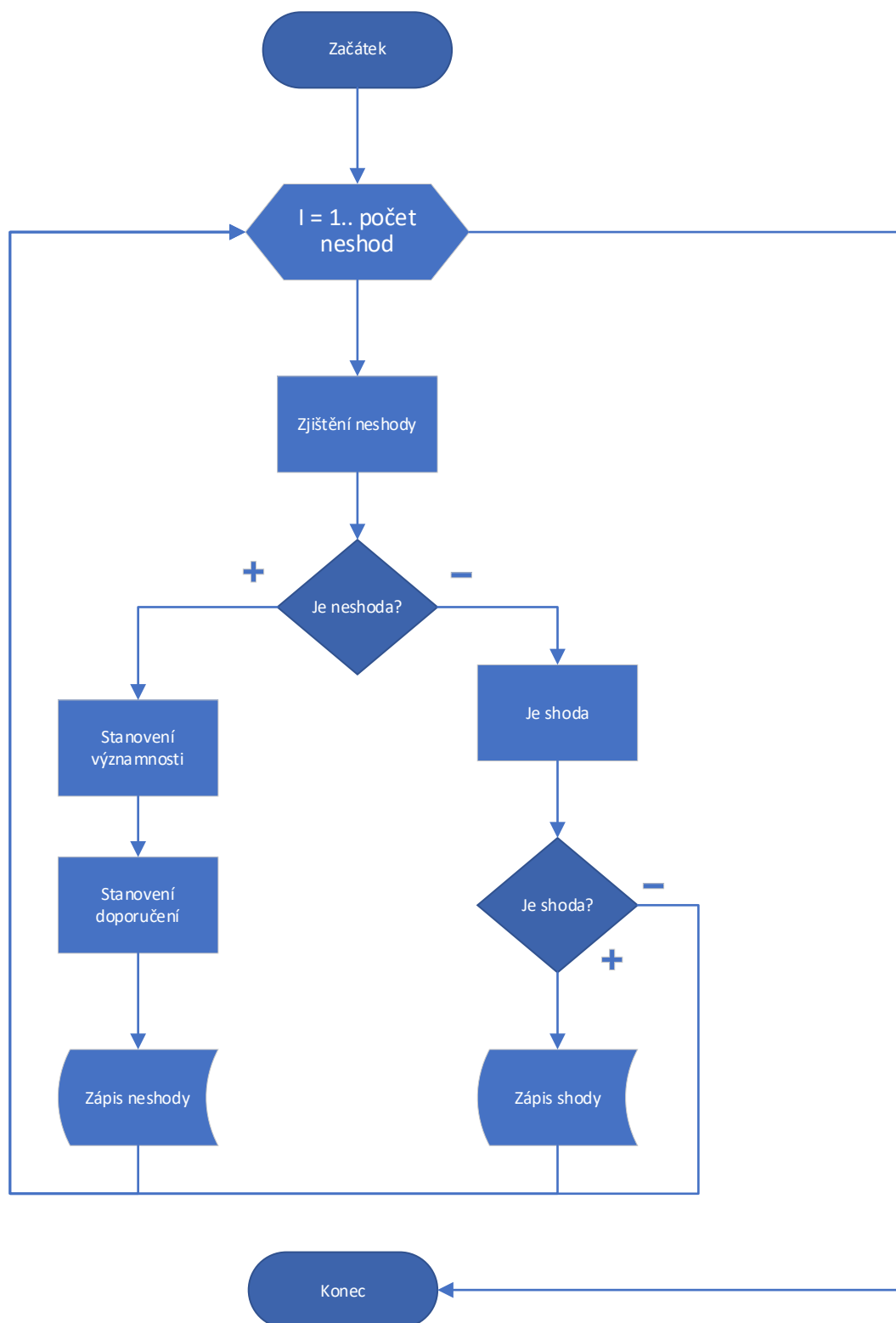
Program také musí umět stanovit podle pravidel, definovaných expertem, významnost zjištěných nedostatků, na škále: vysoká, střední a nízká. Definice významnosti musí být pojata stejně, jako definice neshody, tedy nezávisle na programu. Posledním požadavkem je stanovení případných doporučení k odstranění neshody a jejich zobrazení uživateli při prezentaci výsledků.

Při promýšlení způsobu řešení bylo zjištěno, že tato skutečnost neplatí a shoda s doporučeným stavem je pouze opakem podmínky neshody, protože v případě, že chybí některé odpovědi na testované otázky, nelze určit, zda neshoda či shoda ve skutečnosti nastávají, či ne. Postup jsem shrnul do tabulky

Tabulka č. 2 Zobrazené výsledky při použití neshodných kritérií
(Zdroj: vlastní zpracování)

	neshoda	shoda	nelze určit
vznik	Splnění podmínek odpovědí na kombinaci otázek z dotazníků faktů	Splnění podmínek odpovědí na kombinaci otázek z dotazníků faktů – negace neshody	Chybí některé z odpovědí, které se testují
činnost	Zapsat neshodu Stanovit významnost Zapsat doporučení	Zapsat shodu	

Vlastní výpočet všech nedostatků potom lze znázornit vývojovým diagramem následovně:



Obrázek č. 48 Vývojový diagram identifikace nedostatků
(Zdroj: vlastní zpracování)

3.6 Datové struktury

K realizaci jsou třeba následující datové objekty:

- Tabulka neshod
- Tabulka shod
- Tabulka doporučení
- Tabulka nedostatků
- Tabulka scénářů nedostatků a významnosti

3.6.1 Datový objekt Z_NEDOSTATKY

Datový objekt Z_nedostatky popisuje vlastní nedostatky. Obsahuje název a popis česky i anglicky, dále atribut aktivní, který při hodnotě 1 určuje, že se s nedostatkem pracuje, oblast nedostatku podle číselníků oblastí, id autitu, pro který platí a zda se týká bezpečnosti, nebo ne.

Z_Nedostatky	
 PK id	int(11)
nazev	text
popis	text
nazev_en	text
popis_en	text
aktivni	text
oblast	char(1)
idm	smallint(6)
bezpecnost	tinyint(4)
vaha	decimal(4,1)

Obrázek č. 49 Datový objekt Z_NEDOSTATKY
(Zdroj: vlastní zpracování)

3.6.2 Datový objekt Z_DOPORUCENI

Tento datový objekt definuje rady, doporučení, jak daný nedostatek odstranit. Atributy mají stejný význam a číselníky jako objekt Z_NEDOSTATKY

Z_Doporuceni	
 PK id	int(11)
nazev	text
popis	text
nazev_en	text
popis_en	text
aktivni	int(11)
oblast	char(1)

Obrázek č. 50 Tabulka Z_DOPORUCENI

(Zdroj: vlastní zpracování)

3.6.3 Datový objekt Z_NESHODY / Z_SHODY

Tento datový objekt slouží pro uložení výsledků auditu firem - nalezených neshod a doporučení. Obsahuje cizí klíče na objekty:

Ida – id auditu

Idf – id firmy

Ids – id systému

Idp – id procesu

Idn – id neshody.

riziko - hodnota významnosti nedostatku

Atribut *riziko* není v objektu Z_SHODY, protože zde nemá význam, ale pouze v objektu Z_NESHODY, kde se ukládají nalezené nedostatky.

Atribut riziko, představující významnost rizika, nabývá hodnot:

Klíč	Název česky
2	Nízká
3	Střední
4	Vysoká

Obrázek č. 51 Povolené hodnoty pro atribut riziko

(Zdroj: vlastní zpracování)

Z_Shody	
 PK id	int(11)
idf	int(11)
ida	int(11)
idn	int(11)
ids	int(11)
idp	int(11)
riziko	tinyint(4)

Obrázek č. 52 Tabulka Z_SHODY


(Zdroj: vlastní zpracování)

3.6.4 Datový objekt Z_SCENAR

Tento datový objekt je z pohledu práce nejdůležitější, protože v něm je uložen metajazyk definice výpočtu jednotlivých nedostatků, jejich významnosti a doporučení.

Atribut *idd* určuje typ záznamu a může nabývat hodnot

- 1 – podmínka vzniku neshody
- 2 – významnost neshody
- 3 – doporučení
- 4 – podmínka vzniku shody

Z_Shody	
 PK id	int(11)
idd	int(11)
poradi	int(11)
typ	int(11)
ido	int(11)
volba	set('a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i')

Obrázek č. 53 Z_SCENAR

(Zdroj: vlastní zpracování)

Atribut *poradi* určuje návaznost jednotlivých kroků.

Atribut *typ* definuje příkaz metajazyka a bude vysvětlen dále

Atribut *ido* určuje id otázky, která se vyhodnocuje

Atribut *volba*, typu množina, určuje testované hodnoty odpovědí na otázku

3.7 Definice metajazyka

K identifikaci nedostatku se využívá vyhodnocení odpovědí na otázky dotazníků.

Nedostatek může být definován následovně

POKUD

Odpověď na otázku *i* je hodnota *H1i* nebo *H2i* nebo ..

A současně / Nebo

Odpověď na otázku *j* je hodnota *H1i* nebo *H2j* nebo ..

.....

PAK je nedostatek

Je zřejmé, že vyhodnocení vzniku nedostatku je klasická podmínka if, která ale může být poměrně dlouhá s kombinací logických operátorů AND a OR, a prioritou vyhodnocení výrazu podle závorek.

Metajazyk definice musí tedy obsahovat slova, která lze převést na programovací jazyk. Pro jeho definici byly zavedena následující klíčová slova:

Tabulka č.3 Tabulka Z_AUDITY

(Zdroj: vlastní zpracování)


Klíčové slovo	Popis	Bližší informace
IF	začátek rozhodovacího bloku	
ENDIF	konec rozhodovacího bloku	
(levostranná závorka	
)	pravostranná závorka	
{	levostranná složená závorka	
}	pravostranná složená závorka	
AND	logická spojka a současně	
OR	logická spojka nebo	
OTAZKA	funkce	funkce vrátí hodnotu odpovědi na otázku
RIZIKO	proměnná	do proměnné se nastaví hodnota významnosti nedostatku

DOPORUCENI	proměnná	do proměnné se nastaví id doporučení pro daný nedostatek
-------------------	----------	--

3.8 Formulace nedostatku

Nedostatek musí formulovat expert v dané oblasti. Pro lepší představu je potřeba si tuto skutečnost demonstrovat na příkladu. Je zjišťován nedostatek, zda ve firmě je pro daný informační systém zajištěna uživatelská podpora.


Nedostatek 55

 id	Aktivní	Váha	Metoda	Oblast	Bezpečnost	Název
55	Ano	1.0	Audit systému	Provoz	Ano	Není zajištěna uživatelská podpora

Obrázek č. 54 Příklad nedostatku


(Zdroj: vlastní zpracování)

K tomuto zjištění je potřeba znát odpověď na otázku 142:

 id	Skupina	Oblast	Pořadí	typ	Ukázat	Bezpečnost	Povinná	Otázka
142	Údaje systému	Provoz	15	Jedna odpověď	Ne	Ne	Ne	Kdo poskytuje uživatelskou podporu tohoto informačního systému?

Obrázek č. 55 Informace k otázce 142

(Zdroj: vlastní zpracování)

 Klíč	Volba
a	Informatik z naší firmy
b	Informatik z externí firmy
c	Někdo jiný z naší firmy, kdo není pracovníkem našeho infromatického útvaru
d	Nikdo

Obrázek č. 56 Odpověď k otázce 142


(Zdroj: vlastní zpracování)

Expert dojde například k názoru, že pokud je odpověď na tuto otázku c, nebo d, pak jde o neshodu reálného stavu s doporučením a nedostatek tedy existuje. Současně je přesvědčen, že u malých systémů nebývá tato podpora třeba, takže si zjistí, o jak velký systém jde, u otázky 112.

 id	Skupina	Oblast	Pořadí	typ	Ukázat	Bezpečnost	Povinná	Otázka
112	Údaje systému	Programy	6	Jedna odpověď	Ne	Ne	Ne	Jak je informační systém velký?

Obrázek č. 57 Otázka 112

(Zdroj: vlastní zpracování)

 Klíč	Volba
a	Malý systém nebo program, řádově v tisících Kč
b	Menší systém, v ceně řádově desítky tisíc Kč.
c	Velký systém, v ceně řádově stovky tisíc až miliony Kč.

Obrázek č. 58 Možné odpovědi k předchozí otázce

(Zdroj: vlastní zpracování)

Neshodu tedy definuje následovně:

Pokud je odpověď na otázku 142 c, nebo d a současně je odpověď na otázku 112 b nebo c, pak u firmy je tento nedostatek 55

Pro definici nedostatku byl vytvořen jednoduchý editor v systému.

3.8.1 Definice neshody

V prvním kroku uživatel vybere, zda chce definovat podmínku vzniku neshody, nebo podmínku shody (nedostatek se nevyskytuje), případně význam nedostatku či doporučení pro něj.

Scénář **Není zajištěna uživatelská podpora**

The image shows a web interface for defining a condition. At the top, there is a label 'Skupina' and a dropdown menu labeled 'Podmínka vzniku'. Below this, a larger dropdown menu is open, showing a list of options: 'Podmínka vzniku' (highlighted in blue), 'Riziko', 'Doporučení', and 'Shoda'. To the right of the dropdown menu is an orange button labeled 'Kompilace'.

Obrázek č. 59 Výběrový seznam možností
(Zdroj: vlastní zpracování)

Vlastní scénář je pak popisován pomocí vkládání jednotlivých kroků, které mají být provedeny. Do pole pořadí se doplňuje postupné číslování kroků od 1, druhý sloupec obsahuje příkaz metajazyka, který uživatel vybírá z rozevíracího seznamu. Do třetího pole se píše číslo otázky v případě, že je použit příkaz OTAZKA, a požadované možnosti odpovědi na tuto otázku do sloupce následujícího. Sloupec hodnota slouží k přiřazení hodnoty pro příkazy RIZIKO a DOPORUCENI.

Nastavení podmínky vzniku - P

The image shows a form for setting a condition. It consists of several input fields: 'Pořadí', 'IF' (with a dropdown arrow), 'Otázka', 'IF volba a,b,c,d...', 'hodnota', and a blue button labeled 'Odešli'.

Nastavení podmínky vzniku - P

Pořadí	IF	Otázka	IF voll
1	(
2)		
3	Otázka		
4	Riziko		
	Doporučení		
	AND		
	OR		
	ENDIF		
	{		
	}		

Obrázek č. 60 Výběrový seznam příkazů metajazyka

(Zdroj: vlastní zpracování)

Vyplněný scénář pak vypadá následovně:

Pořadí	IF	Otázka	IF volba a,b,c,d...	hodnota	Odešli
1	IF				
2	Otázka	142	c,d	Kdo poskytuje uživatelskou podporu tohoto informačního systému? Někdo jiný z naší firmy, kdo není pracovníkem našeho informatického útvaru Nikdo	
3	AND				
4	Otázka	112	b,c	Jak je informační systém velký? Menší systém, v ceně řádově desítky tisíc Kč. Velký systém, v ceně řádově stovky tisíc až miliony Kč.	
5	ENDIF				

Obrázek č. 61 Ukázka vyplněného scénáře

(Zdroj: vlastní zpracování)

Kroky jsou číslovány automaticky, pokud je u vkládaného kroku zadáno pořadové číslo, pak se tento krok vloží na uvedený řádek a následující řádky se posunou a přechíslovají. Pokud uživatel použije v rozevřacím seznamu volbu smazat krok a uvede jeho číslo, krok je smazán a následující řádky se opět přechíslovají.

Dále je nutné definovat podmínku shody, tedy stav, kdy se tento nedostatek nevyskytuje. Jak bylo uvedeno v předcházející části práce, neplatí, že shoda je negací neshody, protože může nastat případ, že na některou otázku neznáme odpověď (nebyla

vyplněna v dotazníku), a potom nemůže říct, že nedostatek je, či není. Proto je nutné pro zjištění, že u firmy se tento nedostatek nevyskytuje, formulovat podmínku shody

3.8.2 Definice shody

Shoda se nastavuje stejným způsobem, jako neshoda, vybere se skupina voleb Nastavení shody.

Scénář Není zajištěna uživatelská podpora

Skupina Shoda Kompilace

Nastavení shody - S

Pořadí IF Otázka IF volba a,b,c,d... hodnota Odešli

1	IF				
2	Otázka	142	a,b	Kdo poskytuje uživatelskou podporu tohoto informačního systému? Informatik z naší firmy Informatik z externí firmy	
3	ENDIF				

Obrázek č. 62 Nastavení shody
(Zdroj: vlastní zpracování)

3.8.3 Definice významnosti neshody

Dalším krokem definice neshody je stavení její významnosti, zda tento problém je vážný, či nikoli. V tomto scénáři se do proměnné Riziko nastavuje hodnota z intervalu 2-4 podle číselníku.

Klíč	Název česky
2	Nízká
3	Střední
4	Vysoká

Obrázek č. 63 Úrovně významnosti neshody
(Zdroj: vlastní zpracování)

Obvykle se na počátku scénáře nastaví riziko na hodnotu 4, a pokud jsou splněné podmínky, které toto riziko snižují, tak se snižuje hodnota.

Pokud je v metajazyku použita sekvence IF – ENDIF, tak další řádek uvedený za touto sekvencí se provede v případě, že je podmínka v IF splněna, ale řádek po něm následující už není ve větvi THEN podmínky. Pokud je potřeba, aby se při splnění podmínky provádělo více činností metajazyka, je nutné použít složené závorky metajazyka. V takovém případě by se provedly řádky mezi těmito závorkami ve větvi THEN (podmínka platí). Uvedeném příkladu se řádek 5 provede, pokud je podmínka splněna, a pokud ne, pokračuje se na řádek 6 scénáře (nejsou složené závorky).

Scénář Není zajištěna uživatelská podpora

Skupina

Riziko

Kompilace

Nastavení rizika - R

Pořadí

IF

Otázka

IF volba a,b,c,d...

hodnota

Odešli

1	Riziko			4	
2	IF				
3	Otázka	83	b		Mohla by vaše firma fungovat i bez tohoto informačního systému? <i>Částečně, s většími obtížemi</i>
4	ENDIF				
5	Riziko			3	
6	IF				
7	Otázka	83	c		Mohla by vaše firma fungovat i bez tohoto informačního systému? <i>Ano, s malými nebo žádnými obtížemi</i>
8	ENDIF				
9	Riziko			2	
10	IF				
11	Otázka	101	a,b		Kolik přibližně zaměstnanců má Vaše firma? <i>méně než 10 10-49</i>
12	ENDIF				
13	Riziko			2	

Obrázek č. 64 Nastavení rizika
(Zdroj: vlastní zpracování)

3.8.4 Definice doporučení k nedostatku

Scénář Není zajištěna uživatelská podpora

Skupina: Doporučení [v] Kompilace

Nastavení doporučení- D

Pořadí IF Otázka IF volba a,b,c,d... hodnota Odešli

1	Doporučení	52	Zajistit profesionální uživatelskou podporu
---	------------	----	---

Obrázek č. 65 Menu pro nastavení doporučení
(Zdroj: vlastní zpracování)

Zde je použit příkaz metajazyka Doporučení s id doporučení, které má být k nedostatku zobrazeno.

3.9 Kompilace scénáře

Pro kompilace scénářů byla vytvořena nová funkce v programu. Scénáře se kompilují do php a ukládají se do samostatných souborů, s prefixy

- D - doporučení
- P – nedostatek
- S – shoda
- R - významnost

následovaným id nedostatkem. Pro každý nedostatek se vytváří všechny zkompilované soubory. Zkompilovaný soubor shody se pak použije při kompilaci souboru identifikace nedostatku ve větvi, kdy podmínka neshody neplatí.

Scénář nedostatku

Nejprve se vytvoří php soubor, v komentované příkladu pro nedostatek id 55

```

$id = $this->id;
$skupina = $this->skupina;

$sql = "SELECT * FROM S_SCENARE WHERE idn = '$id' and skupina = '$skupina' order by poradi";
$sv=mysql_query($sql,$db);

    $sl = "kompilace/$skupina";
    $sl .= $id;
    $sl .= ".php";
    $fp = fopen($sl,"w");

    fputs($fp,"<?");
    fputs($fp,"\n");
    fputs($fp,"\n");

```

Obrázek č. 66 PHP kód pro soubor s id 55

(Zdroj: vlastní zpracování)

V prvním průchodu prochází kompilátor celý scénář, a hledá otázky, které budou testovány – klíčové slovo OTAZKA. K těmto otázkám vytvoří řádky, kde do proměnné \$o s indexem čísla otázky se uloží hodnota odpovědi voláním funkce odpoved_otazka(\$ido)

1	IF			
2	Otázka	142	c,d	Kdo poskytuje uživatelskou podporu tohoto informačního systému? Někdo jiný z naší firmy, kdo není pracovníkem našeho informatického útvaru Nikdo
3	AND			
4	Otázka	112	b,c	Jak je informační systém velký? Menší systém, v ceně řádově desítky tisíc Kč. Velký systém, v ceně řádově stovky tisíc až miliony Kč.
5	ENDIF			

```

$o142 = $this->odpoved_otazka(142);
$o112 = $this->odpoved_otazka(112);

$sql = "SELECT * FROM S_SCENARE WHERE idn = '$id' and skupina = '$skupina' and typ = 'd' group by ido order by poradi";
$sv=mysql_query($sql,$db);
while ( $vl=mysql_fetch_object($sv))
{
    $otazka = 'o';
    $otazka .= $vl->ido;
    fputs($fp," $otazka = ");
    fputs($fp,' $this->odpoved_otazka(');
    fputs($fp,$vl->ido);
    fputs($fp,', $idf, $ids, $idp): ');
    fputs($fp,"\n");
}

```

Obrázek č. 67 Ukázka php kódu zpracování scénáře

(Zdroj: vlastní zpracování)

Cílem tohoto kroku je minimalizovat počet dotazů na databázi, pokud je otázka v testu testována na různé možnosti odpovědí a zrychlení práce programu.

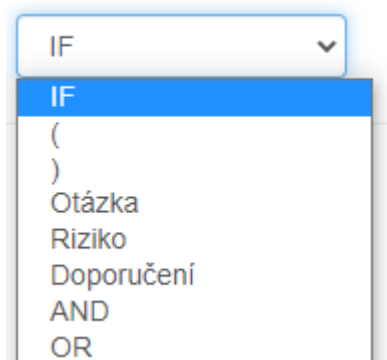
V druhém průchodu prochází kompilátor scénář, a pro definovaná klíčová slova vytváří ekvivalent v programovacím jazyku. Tento kód je pak vytvořen kompilátorem:

```
$o142 = $this->odpoved_otazka(142);
$o112 = $this->odpoved_otazka(112);
if ( ( $o142=='c' or $o142=='d' ) and ( $o112=='b' or $o112=='c' ) )
{
    $index = $this->zapis_nedostatek($idn,$ida,$idf,$ids,$idp,$oblast,$riziko,$bezpecnost);
    if ($ida <> 0) $this->zapis_doporuceni(52,$ida,$idf,$ids,$idp,$oblast,$index);
}
```

Obrázek č. 68 Kód vytvořený kompilátorem

(Zdroj: vlastní zpracování)

Jednotlivá klíčová slova jsou umístěna v seznamu, kde klíčovým slovům odpovídají jednotlivá písmena, například klíčové slovo IF má hodnotu a.



Obrázek č. 69 Seznam klíčových slov

(Zdroj: vlastní zpracování)

Pokud kompilátor najde klíčové slovo IF, vytvoří z něj v php jazyku zápis:

If (

Podmínku uzavírá příkaz Endif, kterému kompilátor přiřadí kód

)

Po vytvoření podmínky vzniku nedostatku se vygenerují řádky pro zápis nedostatku a doporučení pro něj, voláním funkcí zapis_nedostatek a zapis_doporuceni, kde proměnné v hlavičce obsahují identifikátory (id) auditu, firmy, systému, procesu a oblasti. Tyto proměnné se naplňují v hlavní části programu, do kterého se vkládá zkompilovaný kód, jak bude ukázáno na konci této části.

Ukázka části kódu kompilátoru

```
while ( $vl=mysql_fetch_object($v) )
{
    $typ = $vl->typ;
    $otazka = $vl->ido;
    $volby = explode(" ", $vl->volba);
    $hodnota= $vl->hodnota;
    //foreach ($hodn as $h)

    switch ($typ)
    {
        case "a":
            fputs($fp, " if ");
            fputs($fp, " ( ");
            break;

        case "b":
            fputs($fp, " ( ");
            break;

        case "c":
            fputs($fp, " ) ");
            break;
        case "d":

            $pocet = 0;
            if (count($volby) > 1)
            {
                fputs($fp, "(");
            }
            foreach ($volby as $h)
            {
                if ($pocet > 0)
                    fputs($fp, " or ");

                fputs($fp, ' $o' );
                fputs($fp, $otazka);
                fputs($fp, "");
                fputs($fp, "==" );
                fputs($fp, "");
                fputs($fp, $h);
                fputs($fp, "");
                $pocet++;
            }
            if (count($volby) > 1)
            {
                fputs($fp, " ) ");
            }
            break;
    }
}
```

Obrázek č. 70 Ukázka kódu z kompilátoru

(Zdroj: vlastní zpracování)

Po vytvoření části kódu, který zapisuje vznik nedostatku, pokud je splněna podmínka vzniku nedostatku, kompilátor doplní větev ELSE, tedy test a zápis shody, pokud nedostatek nevznikl.

Tento kód definuje uživatel ve scénáři shody

Scénář Není zajištěna uživatelská podpora

Skupina

Nastavení shody - S

Pořadí

1	IF			
2	Otázka	142	a,b	Kdo poskytuje uživatelskou podporu tohoto informačního systému? <i>Informatik z naší firmy</i> <i>Informatik z externí firmy</i>
3	ENDIF			

Obrázek č. 71 Menu pro nastavení shody

(Zdroj: vlastní zpracování)

Dále je uvedena ukázka kompletního zkompilevaného kódu, který zapisuje buď vznik daného nedostatku, nebo potvrzuje, že nedostatek nevznikl. Pokud chybí některé odpovědi na testované otázky, není zapisovaná ani shoda, ani neshoda a daný nedostatek nelze u této firmy vyhodnotit.

Soubor P55.php

```
$o142 = $this->odpoved_otazka(142);
$o112 = $this->odpoved_otazka(112);
if ( ( $o142=='c' or $o142=='d' ) and ( $o112=='b' or $o112=='c' ) )
{
    $index = $this->zapis_nedostatek($idn,$ida,$idf,$ids,$idp,$oblast,$riziko,$bezpecnost);
    if ($ida <> 0) $this->zapis_doporuceni(52,$ida,$idf,$ids,$idp,$oblast,$index);
}
else
{
    if ( ( $o142=='a' or $o142=='b' ) )
    $this->zapis_shodu($idn,$ida,$idf,$ids,$idp,$riziko);
}
```

Obrázek č. 72 Soubor P55.php

(Zdroj: vlastní zpracování)

Tak, jak byl metajazykem definován vznik shody či neshody pro daný nedostatek, definujeme i významnost tohoto nedostatku. Zde se používá klíčové slovo Riziko, které může nabývat hodnoty od 2 do 4.

1	Riziko		4	
2	IF			
3	Otázka	83	b	Mohla by vaše firma fungovat i bez tohoto informačního systému? <i>Částečně, s většími obtížemi</i>
4	ENDIF			
5	Riziko		3	
6	IF			
7	Otázka	83	c	Mohla by vaše firma fungovat i bez tohoto informačního systému? <i>Ano, s malými nebo žádnými obtížemi</i>
8	ENDIF			
9	Riziko		2	
10	IF			
11	Otázka	101	a,b	Kolik přibližně zaměstnanců má Vaše firma? <i>méně než 10</i> <i>10-49</i>
12	ENDIF			
13	Riziko		2	

Obrázek č. 73 Rizika

(Zdroj: vlastní zpracování)

Klíč	Název česky
2	Nízká
3	Střední
4	Vysoká

Obrázek č. 74 Možné hodnoty pro rizika

(Zdroj: vlastní zpracování)

Tento scénář je komplikován stejnou funkcí, analogicky ke vzniku nedostatku.

Stanovená hodnota rizika (významnosti nedostatku pro danou firmu) je pak vložena do proměnné \$riziko, se kterou pracuje funkce zapis_nedostatek

```
$index = $this->zapis_nedostatek($idn,$ida,$idf,$ids,$idp,$oblast,$riziko,$bezpecnost);
```

Soubor R55.php

```
$o83 = $this->odpoved_otazka(83);
$o101 = $this->odpoved_otazka(101);
$riziko = '4';
if ( $o83=='b' )
$riziko = '3';
if ( $o83=='c' )
$riziko = '2';
if ( ( $o101=='a' or $o101=='b' ) )
$riziko = '2';
```

Obrázek č. 75 Soubor R55.php

(Zdroj: vlastní zpracování)

3.10 Finální výpočet

Po dokončení všech čtyřech auditů a vyplnění dotazníků se uživateli zpřístupňuje výpočet a zobrazení výsledků.

Mé audity

Úvod
Mé firmy
Mé systémy
Mé procesy
Mé audity
Mé výsledky
Mé objednávky

Výpočty Nedostatky Průzkumy Efektivnost Bezpečnost

Audity čekající na výpočet

Výpočet je možný, jakmile je dokončen **Audit užítí**. Společně s ním se spočítají audity firmy, systému a procesu, které musí být rovněž dokončeny. **Výsledky** vidíte na dalších záložkách na této straně.

Pokud si v plném přístupu vytvoříte více systémů a procesů, tak pro každý systém nebo proces, u kterého chcete získat výsledky, musíte **vytvořit audit užítí**, který propojuje systém s procesem. Audit užítí se vytváří v menu **Mé audity** tlačítkem **Nový audit**

2074	administrativa
------	----------------

Je nutné provést výpočet výsledků

Výpočet může chvíli trvat. Pro zahájení výpočtu stiskněte prosím tlačítko **Výpočet**

Výpočet

Obrázek č. 76 Menu pro zobrazení výsledků

(Zdroj: vlastní zpracování)

Výpočet probíhá v několika krocích, každý krok je volán dalším refresh stránky po stanovené čase 1 sec.

Mé audity

Úvod
Mé firmy
Mé systémy
Mé procesy
Mé audity
Mé výsledky
Mé objednávky

Výpočty
Nedostatky
Průzkumy
Efektivnost
Bezpečnost

Audity čekající na výpočet

Výpočet je možný, jakmile je dokončen **Audit užítí**. Společně s ním se spočítají audity firmy, systému a procesu, které musí být rovněž dokončeny. **Výsledky** vidíte na dalších záložkách na této straně.

Pokud si v plném přístupu vytvoříte více systémů a procesů, tak pro každý systém nebo proces, u kterého chcete získat výsledky, musíte **vytvořit audit užítí**, který propojuje systém s procesem. Audit užítí se vytváří v menu **Mé audity** | **Mé audity** tlačítkem **Nový audit**

2074	administrativa
------	----------------

Probíhá výpočet výsledků

Po dobu výpočtu prosím neprovádějte žádnou činnost a vyčkejte na konec výpočtů.

Krok 1	Kontrola dat	✓
Krok 2	Zjišťování shody odpovědi	↺
Krok 3	Zjišťování nedostatků	↺

Obrázek č. 77 Průběh hodnocení výsledků
(Zdroj: vlastní zpracování)

Pro každý ze čtyřech auditů se provede výpočet funkcí `vypocet_oblasti_nedostatku`, smažou se předcházející výsledky.

```

$sql = "SELECT * FROM Z_AUDITY where idf = '$idf' and idm=5 ";
$v=mysql_query($sql,$db);
$vl = mysql_fetch_object($v);
if ($vl->vypocet == 0)
{

    $idap=$vl->id;
    $sql = "DELETE FROM Z_VYSLEDKY where ida = '$idap'";
    mysql_query($sql,$db);
    $sql = "DELETE FROM Z_SHODY where ida = '$idap'";
    mysql_query($sql,$db);

    $this->vypocet_oblasti_nedostatku(5,$idap,$idf,0,0);
}

```

Obrázek č. 78 Volání funkce pro výpočet nedostatku
(Zdroj: vlastní zpracování)

Funkce `vypocet_oblasti_nedostatku` pracuje se zkompilovanými soubory výpočtu nedostatků a významnosti, popsány v předchozím odstavci. Zkompilované soubory jsou vkládány do kódu příkazem `INCLUDE` za běhu skriptu, výpočet každého z nedostatků tedy probíhá podle samostatného vloženého podprogramu.

75

```

function vypocet_oblasti_nedostatku($idm,$ida,$idf,$ids,$idp)
{
    $db = mysql_connect(DBTYP,DBLOGIN, DBHESLO) or die;
    mysql_select_db(DBJMENO, $db);mysql_query("SET CHARACTER SET utf8");

    $this->idf = $idf;
    $this->ida = $ida;
    $this->ids = $ids;
    $this->idp = $idp;

    $sql = "SELECT * FROM Z_NEDOSTATKY where aktivni=1 and idm = '$idm'";
    $v=mysql_query($sql,$db);
    while ($vl=mysql_fetch_object($v))
    {
        $idn = $vl->id;

        $bezpecnost = $vl->bezpecnost;
        $oblast = $vl->oblast;
        $riziko = 4;

        $s1 = "kompilace/P";
        $s1 .=$idn;
        $s1 .=".php";
        $s2 = "kompilace/R";
        $s2 .=$idn;
        $s2 .=".php";

        if (is_file("$s1") and is_file("$s2"))
        {
            include("$s2");
            ///////////////////////////////////////////////////
            include("$s1");
            ///////////////////////////////////////////////////
        } // nejsou zkompileovane soubory
    }
}

```

Obrázek č. 79 Funkce výpočtu nedostatku

(Zdroj: vlastní zpracování)

4 EKONOMICKÉ ZHODNOCENÍ

Při stanovení přibližné celkové ceny za nově vytvořené implementované algoritmy pro objektivnější posouzení nedostatků firem je nutno pohlížet i na fakt, že do ceny nelze započítat systém, který existoval dávno před vytvořenými úpravami. Dále platí, že práci, která byla vykonána byť, splňuje zadání a cíl projektu nevytvářel senior programátor se vysokou sazbou za odvedenou práci.

V této části jsou zhodnoceny přínosy a náklady navrženého řešení. Vzhledem k faktu, že systém ZEFIS není zamýšlen jako komerční systém, ale je primárně využíván pro výzkum, sloužící také jako nástroj pro studenty při zpracování bakalářských a diplomových prací a je v naprosté většině případů využíván zdarma, je ekonomické zhodnocení velmi obtížné.

4.1 Náklady

Náklady lze stanovit na základě souhrnného počtu hodin a průměrné mzdy analytika / programátora.

Tabulka č.4 Sumarizace nákladů
(Zdroj: vlastní zpracování)

Činnost	Počet hodin	Průměrná mzda	Skutečné náklady
Analytické práce	20	500 Kč	0 Kč
Úprava databází	15	500 Kč	0 Kč
Programování	50	500 Kč	0 Kč
Celkem:	85	42500 Kč	0 Kč

Pokud by byla práce řešena profesionálním vývojářem, náklady na řešení by se pohybovaly přibližně okolo čtyřiceti tisíc korun. Vzhledem k tomu, že jsem práci dělal zdarma, skutečné finanční náklady jsou nulové.

4.2 Přínosy

U informačních systémů lze přínosy rozdělit do dvou oblastí – přínosy přímé, finančně měřitelné, a přínosy nepřímé, které vyčíslit jednoznačně nelze. Přímé přínosy jsou dány především úsporou času pracovníka v procesu, kterou docílíme zkrácením některé činnosti její náhradou automatizací informačním systémem. Například fakturu lze vytvářet ručně, a pak doba na její vytvoření může být například 10 minut, zatímco při nasazení informačního systému je stejná faktura vytvořena za vteřinu automatizovaně. Docílí se tak úspora času 10 minut na pracovníka a fakturu.

4.2.1 Přímé přínosy

Přímé finanční přínosy vznikají z úspor času v procesu, a dají se spočítat vztahem:

$$P = \text{Počet ušetřených minut} * \text{Počet pracovníků} * \text{Čas} * \text{Průměrná mzda}$$

V případě práce nelze přímé finanční přínosy vyčíslit, protože nespádají do této oblasti.

4.2.2 Nepřímé přínosy

Nepřímé přínosy mají souvislost s informačním systémem, ale nedají se přesně spočítat.

Patří k nim obecně:

- Zlepšení informovanosti pracovníků
- Umožnění nové funkcionality
- Snížení rizika chyby
- Zvýšení bezpečnosti
- Zlepšení rozhodování řídicích pracovníků

Pokud informační systém poskytne například pracovníkovi ve správnou chvíli informaci o platební neschopnosti klienta, může dojít k významné úspoře, pokud se na základě této informace pracovník rozhodne správně, a zboží zákazníkovi nevydá. Tato úspora ale má proměnlivou hodnotu a závisí na faktu, zda pracovník získanou informaci správně využije.

V rámci práce jsou přínosy především v oblasti umožnění nové funkcionality a získání nových informací ze získaných dat.

- Informace o možných nedostatecích
- Významnost těchto nedostatků
- Porovnání stavu s ostatními firmami

Tyto informace z části uživatel systému ví, ale nemůže je bez provedení auditu posoudit a ověřit. Přínosy řešení z pohledu uživatele systému vedou především k zlepšení současného stavu užívání informačních systémů, vyhnutí se chybám oproti best-practises a standartním postupům a zlepšení úrovně bezpečnosti. Pokud firma zlepší stávající stav, může ji to přinést významné úspory – například se vyhne velké ztrátě ze zničených dat, pokud nebyla předtím zálohována.

Z pohledu provozovatele systému je přínosem zlepšení funkcionality a získání nových informací pro vlastní výzkum v oblasti efektivity a efektivnosti informačních systémů.

5 ZÁVĚR

Cílem bakalářské práce bylo rozšířit informační systém Zefis o možnost zjišťovat nedostatky firem, jejich informačních systémů a procesů. Byl dán požadavek, že nově musí systém umožňovat identifikovat nedostatek na základě kombinace odpovědí na libovolné otázky dotazníků. Nedostatek definuje v systému ZEFIS expert, který popisuje tento nedostatek pomocí metajazyka mimo vlastní program tak, aby bylo možné popisy nedostatků vytvářet a měnit bez nutnosti měnit program.

Takto popsané nedostatky pak umí systém diagnostikovat při zpracování výpočtu auditů na reálných datech

Dílčím cílem práce bylo také umožnit vyhledávání odpovědí v dotaznících v závislosti na odpovědi na jiné otázky, například zjistit, u kolika malých firem v oblasti strojírenství není ustanoven manažer informačních systémů. Tento cíl vyžadoval změnu původních datových struktur a způsobu vyhledávání v databázi pomocí SQL dotazů a je primárně využíván pro analýzy dat provozovatelem systému.

Lze konstatovat, že oba požadované cíle byly v práci splněny, a informační systém ZEFIS v současné verzi již s touto funkcionalitou pracuje.

SEZNAM POUŽITÝCH ZDROJŮ

- (1) CONOLLY, Thomas, Carolyn E. BEGG a Richard HOLOWCZAK. Mistrovství - databáze: profesionální průvodce tvorbou efektivních databází. 1. vyd. Brno: Computer Press, 2009. 584 s. ISBN 978-80-251-2328-7.
- (2) GILMORE, W. J. Velká kniha PHP a MySQL 5: kompendium znalostí pro začátečníky i profesionály. 1. vyd. [i.e. 2. vyd.]. Brno: Zoner Press, 2007. 864 s. ISBN 80-86815-53-6.
- (3) GÁLA, Libor, Jan POUR a Zuzana ŠEDIVÁ. Podniková informatika. 3. vyd. Praha: Grada Publishing, 2015. 240 s. ISBN 978-80-247-5457-4.
- (4) MOLNÁR, Zdeněk. Efektivnost informačních systémů. 2. vyd. Praha: Grada, 2001. 179 s. ISBN 80-2470-087-5.
- (5) SODOMKA, Petr a Hana KLČOVÁ. Informační systémy v podnikové praxi. 2. vyd. Brno: Computer Press, 2010. 504 s. ISBN 978-80-251-2878-7.
- (6) JANOVSKEÝ, Dušan. HTML příručka, přehled HTML tagů. *Jak psát web, návod na html stránky* [online]. ©1999 - 2021 [cit. 12. 03. 2021]. Dostupné z <https://www.jakpsatweb.cz/html/>
- (7) JANOVSKEÝ, Dušan. Nejdůležitější tagy. *Jak psát web, návod na html stránky* [online]. ©1999 - 2021 [cit. 12. 03. 2021]. Dostupné z <https://www.jakpsatweb.cz/html/nejdulezitejsi.html>
- (8) ITNETWORKS. Lekce 1 - Úvod do HTML a váš první web. *itnetwork.cz - Ajtácká sociální síť a materiálová základna pro C#, Java, PHP, HTML, CSS, JavaScript a další.* [online]. © 2021 [cit. 13.03.2021]. Dostupné z: <https://www.itnetwork.cz/html-css/webove-stranky/jak-psat-moderni-web-html-tutorial-uvod-do-html>
- (9) W3SCHOOLS.COM. Introduction to HTML. *W3Schools Online Web Tutorials* [online]. © 1999-2016 [cit. 12.03.2021]. Dostupné z: https://www.w3schools.com/html/html_intro.asp
- (10) MENCL, Michal. Základní kurz 1: Úvod. *Péhápkó: Učebnice PHP* [online]. [cit. 18.03.2021] Dostupné z: <http://www.pehapko.cz/zakladni-kurz/1-uvod>

- (11) W3SCHOOLS.COM. PHP Syntax. *W3Schools Online Web Tutorials* [online]. © 1999-2016 [cit. 12.03.2021]. Dostupné z:
https://www.w3schools.com/php/php_syntax.asp
- (12) MENCL, Michal. Základní kurz 4: Základy syntaxe. *Péhápkó: Učebnice PHP* [online]. [cit. 18.03.2021] Dostupné z: <http://www.pehanko.cz/zakladni-kurz/4-zaklady-syntaxe>
- (13) W3SCHOOLS.COM. PHP Data Types. *W3Schools Online Web Tutorials* [online]. © 1999-2016 [cit. 19.03.2021]. Dostupné z:
https://www.w3schools.com/php/php_datatypes.asp
- (14) W3SCHOOLS.COM. PHP Functions. *W3Schools Online Web Tutorials* [online]. © 1999-2016 [cit. 19.03.2021]. Dostupné z:
https://www.w3schools.com/php/php_datatypes.asp
- (15) W3SCHOOLS.COM. SQL Joins. *W3Schools Online Web Tutorials* [online]. © 1999-2016 [cit. 24.03.2021]. Dostupné z:
https://www.w3schools.com/sql/sql_join.asp
- (16) W3SCHOOLS.COM. SQL Data Types for MySQL, SQL Server, and MS Access. *W3Schools Online Web Tutorials* [online]. © 1999-2016 [cit. 26.03.2021]. Dostupné z: https://www.w3schools.com/sql/sql_datatypes.asp

SEZNAM POUŽITÝCH OBRÁZKŮ

Obrázek č. 1: Příklad jednořádkového komentáře	14
Obrázek č. 2: Ukázkový příklad použití příkazu if	16
Obrázek č. 3: Použití větve else	16
Obrázek č. 4: Deklarace řetězce	17
Obrázek č. 5: Ukázka for cyklu	18
Obrázek č. 6: Příklad funkce s parametry	19
Obrázek č. 7: Příklad funkce bez vstupních parametrů	19
Obrázek č. 8: Příklad třídy	20
Obrázek č. 9: Metoda k předchozí třídě	21
Obrázek č. 10: Znázorněné průniky dat na základě spojení tabulek	24
Obrázek č. 11 Tabulka Z_FIRMY	31
Obrázek č. 12 Tabulka Z_SYSTEMY	32
Obrázek č. 13 Tabulka Z_SYSTEMY	32
Obrázek č. 14 Tabulka Z_AUDITY	34
Obrázek č. 15 Tabulka Z_RESPONDENTI (Zdroj: vlastní zpracování)	35
Obrázek č. 16 Tabulka Z_ODPOVEDI	36
Obrázek č. 17 Tabulka Z_OTAZKY	37
Obrázek č. 18 Tabulka Z_MOZNOSTI	38
Obrázek č. 19 Příklad otázky s dvěma možnostmi	38
Obrázek č. 20 Ukázka auditů v systému	39
Obrázek č. 21 Dotazník k auditu	39
Obrázek č. 22 Otázka 46	40
Obrázek č. 23 Příklad otázky v systému	41
Obrázek č. 24 Příklad otázky v systému s vyšším počtem možných odpovědí	41
Obrázek č. 25 Výsledek odpovědi na otázku	42
Obrázek č. 26 Pohled z hlediska firmy	43
Obrázek č. 27 Pohled z hlediska systému	43
Obrázek č. 28 Pohled z hlediska procesu	43
Obrázek č. 29 Pohled z hlediska užití	44
Obrázek č. 30 ER model pro zjišťování nedostatků	44
Obrázek č. 31 ER model úpravy struktury odpovědi na otázky	45
Obrázek č. 32 Tabulka Z_ODPOVEDI	45
Obrázek č. 33 Příklad otázky s jednou možnou odpovědí	46
Obrázek č. 34 Výčet možných odpovědi na otázku	47
Obrázek č. 35 Doplněný parametr	47
Obrázek č. 36 Uložené odpovědi z dotazníků	47
Obrázek č. 37 Příklad otázky s více možnými odpověďmi	48
Obrázek č. 38 Výčet možných odpovědi na otázku	48
Obrázek č. 39 Doplněný atribut	49
Obrázek č. 40 Uložené odpovědi z dotazníků	49
Obrázek č. 41 Doprogramovaná tlačítka na přidání atributu	49
Obrázek č. 42 Upravený kód v modulu kontroly dotazníku	50

Obrázek č. 43 Příklad otázky	51
Obrázek č. 44 Ukázka informace o stavu navolené podmínky	52
Obrázek č. 45 Zobrazené výsledky při použití jednoho kritéria	52
Obrázek č. 46 Zobrazené výsledky při použití více kritérií	53
Obrázek č. 47 Zobrazené výsledky při použití neshodných kritérií	53
Obrázek č. 48 Vývojový diagram identifikace nedostatků (Zdroj: vlastní zpracování) ..	56
Obrázek č. 49 Datový objekt Z_NEDOSTATKY	57
Obrázek č. 50 Tabulka Z_DOPORUCENI	58
Obrázek č. 51 Povolené hodnoty pro atribut riziko	59
Obrázek č. 52 Tabulka Z_SHODY	59
Obrázek č. 53 Z_SCENAR	60
Obrázek č. 54 Příklad nedostatku	62
Obrázek č. 55 Informace k otázce 142	62
Obrázek č. 56 Odpověď k otázce 142	63
Obrázek č. 57 Otázka 112	63
Obrázek č. 58 Možné odpovědi k předchozí otázce	63
Obrázek č. 59 Výběrový seznam možností	64
Obrázek č. 60 Výběrový seznam příkazů metajazyka	65
Obrázek č. 61 Ukázka vyplněného scénáře	65
Obrázek č. 62 Nastavení shody	66
Obrázek č. 63 Úrovně významnosti neshody	66
Obrázek č. 64 Nastavení rizika	67
Obrázek č. 65 Menu pro nastavení doporučení	68
Obrázek č. 66 PHP kód pro soubor s id 55	69
Obrázek č. 67 Ukázka php kódu zpracování scénáře	69
Obrázek č. 68 Kód vytvořený kompilátorem	70
Obrázek č. 69 Seznam klíčových slov	70
Obrázek č. 70 Ukázka kódu z kompilátoru	71
Obrázek č. 71 Menu pro nastavení shody	72
Obrázek č. 72 Soubor P55.php	72
Obrázek č. 73 Rizika	73
Obrázek č. 74 Možné hodnoty pro rizika	73
Obrázek č. 75 Soubor R55.php	74
Obrázek č. 76 Menu pro zobrazení výsledků	74
Obrázek č. 77 Průběh hodnocení výsledků	75
Obrázek č. 78 Volání funkce pro výpočet nedostatku	75
Obrázek č. 79 Funkce výpočtu nedostatku	76

SEZNAM POUŽITÝCH TABULEK

Tabulka č.1 Příklady použitých datových typů pro SQL.....	26
Tabulka č.2 Zobrazené výsledky při použití neshodných kritérií.....	55
Tabulka č.3 Tabulka Z_AUDITY	61
Tabulka č.4 Sumarizace nákladů	77